

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Estudio de la precisión de los dispositivos de *eye-tracking* para la
evaluación de patrones de lectura**

Víctor Díaz Herráez
Tutor: Pablo Varona Martínez

JUNIO - JULIO 2017

Estudio de la precisión de los dispositivos de *eye-tracking* para la evaluación de patrones de lectura

AUTOR: Víctor Díaz Herráez
TUTOR: Pablo Varona Martínez

Grupo de Neurocomputación Biológica
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio - Julio de 2017

Resumen (castellano)

El seguimiento de la actividad ocular es, en la actualidad, una de las tecnologías en auge que se puede usar en una gran diversidad de sectores y con finalidades muy distintas. Actualmente existe una gran variedad de dispositivos de seguimiento ocular o *eye-trackers*. Estos aparatos registran la actividad ocular mediante el uso de la tecnología de iluminación infrarroja que se refleja en el ojo, y se emplean en una gran variedad de campos, tales como psicología, medicina, marketing o, más recientemente, en el sector de los videojuegos.

En este trabajo se aborda la comparación de la precisión de dos dispositivos de distintas compañías en la medición de la actividad ocular en el proceso de lectura de textos, para cuantificar la diferencia entre dispositivos de bajo coste (como es el caso del dispositivo de TheEyeTribe) y de gama alta (como los dispositivos de Tobii). Esta comparativa es de utilidad en el diseño de aplicaciones de recomendación de diagnóstico de dislexia y déficits de atención.

La aplicación parte de un desarrollo previo y se ha adaptado para ser utilizada en cualquier dispositivo con un SDK compatible con .Net y, asimismo, graba el audio durante la prueba, para facilitar la evaluación de patrones de lectura. Además de proporcionar los datos del seguimiento ocular, genera, en un fichero adjunto en la carpeta de resultados, información adicional a la prueba que permite la evaluación cuantitativa de la precisión del *eye-tracker* en las tareas de lectura de textos, así como su comparación con otras pruebas.

Para validar la aplicación, se han realizado una serie de pruebas con voluntarios cuyos resultados han sido analizados y comparados. Finalmente, en este documento se exponen las conclusiones obtenidas y los posibles trabajos futuros.

Abstract (English)

The monitoring of the ocular activity is, now days, one of the emerging technologies in different applications, since it can be used in a great diversity of sectors and with very different purposes. There is a large variety of eye-trackers. These devices record ocular activities using infrared technology (in some cases accompanied by images), and are used in a wide variety of fields such as psychology, medicine, marketing or, more recently, in the field of videogames.

This work attempts to compare the accuracy of devices from different companies while measuring the ocular activity during the process of reading texts, quantifying the difference between low-end devices (such as the Eye Tribe device) and high-end devices (such as Tobii devices). This comparison is useful on the design of applications for the diagnose of dyslexia and attention deficit.

The application is part of a previous development and it has been adapted to be used by any device with a SDK compatible with .Net and, moreover, it records the audio during the tests, as it makes easier the evaluation of reading patterns. In addition, when generating the results, an extra file is computed and attached into the results folder, containing supplementary information to the test that allows the quantitative evaluation of the eye-trackers precision during the reading process, as well as the comparison with other tests.

To validate the application, a series of tests have been carried out with volunteers whose results have been analyzed and compared. Finally, this document presents the conclusions obtained and exposing possible future work from this project.

Palabras clave (castellano)

Rastreadores oculares, seguimiento de ojos, actividad ocular, precisión, evaluación de patrones de lectura, precisión en el seguimiento ocular

Keywords (inglés)

Eye tracker, eye tracking, ocular activity, precision, reading pattern evaluation, eye-tracking precision.

Agradecimientos

En primer lugar, quiero expresar mi agradecimiento a Pablo Varona por permitir que pudiera hacer el TFG con él y por ayudarme cuando lo he necesitado.

Asimismo, quiero agradecer a Ruth Campos y Pablo Campo por darme acceso al dispositivo de Tobii que tienen en la Facultad de Psicología.

También quiero dar las gracias a toda mi familia y amigos, que han estado siempre apoyandome en los momentos difíciles.

Y, por supuesto, a todos los que se han prestado voluntarios para realizar las pruebas, ese gran equipo de Hava-Soft, familiares y amigos, pues sin ellos no sería lo mismo.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Organización de la memoria.....	1
2	Estado del arte	3
2.1	Historia del Eye-Tracking	3
2.2	Eye-Tracker	4
2.3	Campos en los que se emplea la tecnología eye-tracking	8
2.4	Precisión	9
3	Diseño.....	11
3.1	Dispositivos	11
3.1.1	Colocación de los dispositivos	11
3.1.2	Calibración.....	12
3.1.3	Datos proporcionados	13
3.2	Usuarios	13
3.3	Aplicación.....	14
3.3.1	Universalidad de uso.....	14
3.3.2	Distribución del código	14
3.4	Ficheros de lectura.....	15
3.5	Grabación del audio.....	15
3.6	Calibración.....	15
3.7	Resultados.....	15
4	Desarrollo	17
4.1	Herramientas usadas	17
4.2	Aplicación.....	17
4.2.1	Registro de patrones	18
4.2.2	Generación de Resultados.....	19
4.2.2.1	Generación de los resultados multimedia.....	19
4.2.2.2	Cálculo de la precisión	20
4.3	Interfaz Gráfica.....	20
5	Integración, pruebas y resultados	25
5.1	Objetivos de la evaluación.....	25
5.2	La prueba	25
5.2.1	Textos para la lectura.....	25
5.3	Resultados.....	26
5.3.1	Usuario 1.....	26
5.3.1.1	Texto 1.....	27
5.3.1.2	Texto 2.....	27
5.3.2	Usuario 3.....	28
5.3.2.1	Texto 1.....	28
5.3.2.2	Texto 2.....	28
5.3.3	Usuario 7.....	29
5.3.3.1	Texto 1.....	29
5.3.3.1	Texto 2.....	30
5.3.4	Usuario 10.....	30
5.3.4.1	Texto 1.....	31
5.3.4.2	Texto 2.....	31

.....	31
5.3.5 Usuario 11.....	32
5.3.5.1 Texto 1.....	32
5.3.5.2 Texto 2.....	32
5.4 Análisis de la precisión.....	33
6 Conclusiones y trabajo futuro.....	35
6.1 Conclusiones.....	35
6.2 Trabajo futuro.....	36
Referencias.....	37
Glosario.....	39
Anexos.....	- 1 -
A Manual de instalación.....	- 1 -
B Diagrama de Clases.....	- 3 -
C Calibración del dispositivo de TheEyeTribe.....	- 5 -
D Calibración del dispositivo de Tobii.....	- 7 -
E Código.....	- 9 -
F Cálculo de la precisión y la distancia media de fallo.....	- 15 -

INDICE DE FIGURAS

FIGURA	2-1:	EYE	TRACKER	DE	YARBUS	
		HTTPS://UPLOAD.WIKIMEDIA.ORG/WIKIPEDIA/COMMONS/A/A8/YARBUS_EYE_TRACKER.JPG . 3				
FIGURA	2-2:	TOBII	PRO	GLASSES		2
		HTTP://WWW.TOBIIPRO.COM/IMAGEVAULT/PUBLISHEDMEDIA/JN0KNQ8UNOTEXK30Z8B5/TOBIIPRO_GLASSES_2_EYE_TRACKER_SIDE_3_1.JPG 5				
FIGURA	2-3:	TOBII			X2-30	
		HTTP://WWW.TOBIIPRO.COM/IMAGEVAULT/PUBLISHEDMEDIA/99ZXPC4H39MU8SFVBU44/TOBIIPRO_X2_EYE_TRACKER_3_1.JPG 5				
FIGURA	2-4:	THE	EYE	TRIBE	TRACKER	
		HTTPS://S3.EU-CENTRAL-1.AMAZONAWS.COM/THEEYETRIBE.COM/THEEYETRIBE.COM/LH5.GOOGLEUSERCONTENT.COM/DEMO-MODE_MINIMUZWD.JPG 6				
FIGURA	2-5:	TOBII	PRO		Tx300	
		HTTP://WWW.TOBIIPRO.COM/IMAGEVAULT/PUBLISHEDMEDIA/NFK0U0RXGR964YMHK7IW/TOBIIPRO_TX300_EYE_TRACKER_3_1.JPG 6				
FIGURA	2-6:	HEATMAP	HACIENDO	LA	COMPRA	
		HTTPS://CANADASBESTSTOREFIXTURESBLOG.FILES.WORDPRESS.COM/2013/07/HEATMAP.PNG 8				
FIGURA	2-7:	HEATMAP	PÁGINA	WEB		
		HTTPS://WWW.GODADDY.COM/GARAGE/WP-CONTENT/UPLOADS/2017/03/HEAT-MAP-EXAMPLES-EYE-TRACKING.JPG 8				

FIGURA 2-8: EYE TRACKING PARA DETECCIÓN DE PATRONES DE LECTURA HTTPS://INSTAGLOSS.COM/STATIC/READING_SKIMMING.JPG.....	9
FIGURA 2-9: PRECISIÓN DE LA MIRADA HTTP://WWW.TOBII.PRO.COM/IMAGEVAULT/PUBLISHEDMEDIA/PU6JY5UF9WPT7Y3OG1K4/ACCURACY___PRECISION.PNG.....	9
FIGURA 2-10: MUESTREO PRECISIÓN DE LA MIRADA HTTP://WWW.TOBII.PRO.COM/SITEASSETS/TOBII-PRO/LEARN-AND-SUPPORT/USE/WHAT-AFFECTS-THE-PERFORMANCE-OF-AN-EYE-TRACKER/TOBII-TEST-SPECIFICATIONS-ACCURACY-AND-PRECISION-TEST-METHOD.PDF/?v=2.1.1.....	10
FIGURA 3-1: COLOCACIÓN DISPOSITIVO TOBII HTTPS://WWW.TOBII.PRO.COM/IMAGEVAULT/PUBLISHEDMEDIA/S48HJ6KT1J174ALMTF4S/TX300_SETUP_2_1.JPG.....	11
FIGURA 3-2: COLOCACIÓN DISPOSITIVO EYE TRIBE HTTPS://S3.EU-CENTRAL-1.AMAZONAWS.COM/THEEYETRIBE.COM/THEEYETRIBE.COM/DEV/GENERAL/INDEX.HTML ...	12
FIGURA 3-3: CALIBRACIÓN TOBII	12
FIGURA 3-4: CALIBRACIÓN EYE TRIBE.....	13
FIGURA 4-1: ERROR "NO HAY MICRÓFONO CONECTADO"	21
FIGURA 4-2: MENÚ PRINCIPAL	21
FIGURA 4-3: ERROR PASOS PREVIOS AL REGISTRO NO REALIZADOS.....	21
FIGURA 4-4: REGISTRO DEL USUARIO	22
FIGURA 4-5: ERROR EDAD INTRODUCIDA NO VÁLIDA.....	22
FIGURA 4-6: ERROR FALTAN DATOS	22
FIGURA 4-7: CONFIGURACIÓN DE LA PRUEBA.....	22
FIGURA 4-8: MENÚ DE GENERACIÓN DE RESULTADOS.....	23
FIGURA 4-9: GENERACIÓN DE LOS RESULTADOS DE UNA PRUEBA	23
FIGURA 5-1: ERROR EN EL PROCESO DE CALIBRACIÓN DE EYE TRIBE	26
FIGURA 5-2: REGISTRO USUARIO 1 TEXTO 1 OJO DERECHO EYE TRIBE.....	27
FIGURA 5-3: REGISTRO USUARIO 1 TEXTO 1 OJO IZQUIERDO EYE TRIBE	27
FIGURA 5-4: REGISTRO USUARIO 1 TEXTO 1 OJO DERECHO TOBII	27
FIGURA 5-5: REGISTRO USUARIO 1 TEXTO 1 OJO IZQUIERDO TOBII.....	27
FIGURA 5-6: REGISTRO USUARIO 1 TEXTO 2 OJO DERECHO EYE TRIBE.....	27

FIGURA 5-7: REGISTRO USUARIO 1 TEXTO 2 OJO IZQUIERDO EYEtribe	27
FIGURA 5-8: REGISTRO USUARIO 1 TEXTO 2 OJO IZQUIERDO TOBII.....	28
FIGURA 5-9: REGISTRO USUARIO 1 TEXTO 2 OJO IZQUIERDO TOBII.....	28
FIGURA 5-10: RESULTADOS USUARIO 3 TEXTO 1 OJO DERECHO EYEtribe.....	28
FIGURA 5-11: RESULTADOS USUARIO 3 TEXTO 1 OJO IZQUIERDO EYEtribe	28
FIGURA 5-12: RESULTADOS USUARIO 3 TEXTO 1 OJO DERECHO TOBII	28
FIGURA 5-13: RESULTADOS USUARIO 3 TEXTO 1 OJO IZQUIERDO TOBII.....	28
FIGURA 5-14: RESULTADOS USUARIO 3 TEXTO 2 OJO IZQUIERDO EYEtribe	29
FIGURA 5-15: RESULTADOS USUARIO 3 TEXTO 2 OJO DERECHO EYEtribe.....	29
FIGURA 5-16: RESULTADOS USUARIO 3 TEXTO 2 OJO IZQUIERDO TOBII.....	29
FIGURA 5-17: RESULTADOS USUARIO 3 TEXTO 2 OJO DERECHO TOBII	29
FIGURA 5-18: RESULTADOS USUARIO 7 TEXTO 1 OJO IZQUIERDO EYEtribe	29
FIGURA 5-19: RESULTADOS USUARIO 7 TEXTO 1 OJO DERECHO EYEtribe.....	29
FIGURA 5-20: RESULTADOS USUARIO 7 TEXTO 1 OJO DERECHO TOBII	30
FIGURA 5-21: RESULTADOS USUARIO 7 TEXTO 1 OJO DERECHO TOBII	30
FIGURA 5-22: RESULTADOS USUARIO 7 TEXTO 2 OJO DERECHO EYEtribe.....	30
FIGURA 5-23: RESULTADOS USUARIO 7 TEXTO 2 OJO IZQUIERDO EYEtribe	30
FIGURA 5-24: RESULTADOS USUARIO 7 TEXTO 2 OJO IZQUIERDO TOBII.....	30
FIGURA 5-25: RESULTADOS USUARIO 7 TEXTO 2 OJO DERECHO TOBII	30
FIGURA 5-26: RESULTADOS USUARIO 10 TEXTO 1 OJO DERECHO EYEtribe.....	31
FIGURA 5-27: RESULTADOS USUARIO 10 TEXTO 1 OJO IZQUIERDO EYEtribe	31
FIGURA 5-28: RESULTADOS USUARIO 10 TEXTO 1 OJO DERECHO TOBII	31
FIGURA 5-29: RESULTADOS USUARIO 10 TEXTO 1 OJO IZQUIERDO TOBII.....	31
FIGURA 5-30: RESULTADOS USUARIO 10 TEXTO 2 OJO DERECHO EYEtribe.....	31
FIGURA 5-31: RESULTADOS USUARIO 10 TEXTO 2 OJO IZQUIERDO EYEtribe	31
FIGURA 5-32: RESULTADOS USUARIO 10 TEXTO 2 OJO IZQUIERDO TOBII.....	32

FIGURA 5-33: RESULTADOS USUARIO 10 TEXTO 2 OJO DERECHO TOBII	32
FIGURA 5-34: RESULTADOS USUARIO 11 TEXTO 1 OJO DERECHO EYEtribe.....	32
FIGURA 5-35: RESULTADOS USUARIO 11 TEXTO 1 OJO IZQUIERDO EYEtribe	32
FIGURA 5-36: RESULTADOS USUARIO 11 TEXTO 1 OJO DERECHO TOBII	32
FIGURA 5-37: RESULTADOS USUARIO 11 TEXTO 1 OJO IZQUIERDO TOBII.....	32
FIGURA 5-38: RESULTADOS USUARIO 11 TEXTO 2 OJO DERECHO EYEtribe.....	33
FIGURA 5-39: RESULTADOS USUARIO 11 TEXTO 2 OJO IZQUIERDO EYEtribe	33
FIGURA 5-40: RESULTADOS USUARIO 11 TEXTO 2 OJO IZQUIERDO TOBII.....	33
FIGURA 5-41: RESULTADOS USUARIO 11 TEXTO 2 OJO DERECHO TOBII	33
FIGURA 0-1: DIAGRAMA DE CLASES DE LA APLICACIÓN.....	- 3 -
FIGURA 0-2: OJOS NO DETECTADOS EYEtribe FIGURA 0-3: OJOS DETECTADOS EYEtribe .	- 5 -
FIGURA 0-4: RESULTADO CALIBRACIÓN	- 5 -
FIGURA 0-5: PANEL DE CONFIGURACIÓN DE LA APLICACIÓN DE EYEtribe.....	- 6 -
FIGURA 0-6: APLICACIÓN CALIBRACIÓN TOBII.....	- 7 -
FIGURA 0-7: TOBII NO RECONOCE OJOS.	- 7 -
FIGURA 0-8: TOBII RECONOCIENDO OJOS.....	- 8 -
FIGURA 0-9: RESULTADOS CALIBRACIÓN TOBII	- 8 -
FIGURA 0-10: CÓDIGO INICIALIZACIÓN Y COMIENZO DEL REGISTRO.	- 9 -
FIGURA 0-11: CÓDIGO DEL REGISTRO DE LA ACTIVIDAD OCULAR I	- 9 -
FIGURA 0-12: CÓDIGO DEL REGISTRO DE LA ACTIVIDAD OCULAR II	- 10 -
FIGURA 0-13: OBTENCIÓN DATOS EYEtribe.....	- 10 -
FIGURA 0-14: OBTENCIÓN DATOS TOBII	- 11 -
FIGURA 0-15: FONDO DE LOS GRÁFICOS	- 11 -
FIGURA 0-16: CÓDIGO GENERACIÓN DE RESULTADOS "LEFT_FULL.PNG" Y "RIGHT_FULL.PNG" .	- 11 -
FIGURA 0-17: CÓDIGO GENERACIÓN DE LOS RESULTADOS "LEFT.PNG" Y "RIGHT.PNG"	- 11 -
FIGURA 0-18: CÓDIGO GENERACIÓN DE LOS RESULTADOS "LEFT_COLOR.PNG" Y "RIGHT_COLOR.PNG"	- 11 -

FIGURA 0-19: CÓDIGO GENERACIÓN DE LOS RESULTADOS "LEFT_PUPIL.PNG" Y "RIGHT_PUPIL.PNG"	- 12 -
FIGURA 0-20: CÓDIGO GENERACIÓN DE LOS GRÁFICOS USADOS PARA LOS VIDEOS "LEFT_MOVIE.MPEG" Y "RIGHT_MOVIE.MPEG"	- 12 -
FIGURA 0-21: CÓDIGO COMPROBACIÓN DE DESPLAZAMIENTO VERTICAL	- 12 -
FIGURA 0-22: CÓDIGO LIMPIEZA DE REGISTROS PREVIOS AL COMIENZO DE LA LECTURA	- 12 -
FIGURA 0-23: CÓDIGO CÁLCULO DE LA PRECISIÓN	- 13 -
FIGURA 0-24: TEXTO DE EJEMPLO	- 15 -
FIGURA 0-25: TEXTO CON LAS LINEAS DELIMITADAS	- 15 -
FIGURA 0-26: CÁLCULO DE LA DISTANCIA	- 16 -
FIGURA 0-27: TEXTO CON LÍNEAS Y PUNTOS MARCADOS	- 17 -

INDICE DE TABLAS

TABLA 2-1: ACTIVIDADES REGISTRADAS POR LOS EYE TRACKERS	4
TABLA 2-2: COMPARATIVA DE LOS SEGUIDORES DE MIRADA USADOS	7
TABLA 3-1: DATOS PROPORCIONADOS POR LOS SDK	13
TABLA 5-1: RESULTADOS TEXTO 1 OJO IZQUIERDO	34
TABLA 5-2: RESULTADOS TEXTO 1 OJO DERECHO	34
TABLA 5-3: RESULTADOS TEXTO 2 OJO IZQUIERDO	34
TABLA 5-4: RESULTADOS TEXTO 2 OJO DERECHO	34

1 Introducción

1.1 Motivación

En la actualidad hay una gran variedad de dispositivos de seguimiento ocular, los cuales se emplean en distintos campos como la medicina, psicología y la industria de los videojuegos, entre otros. Entre estos dispositivos puede llegar a haber una diferencia importante en los precios. Por ejemplo, tenemos a la marca TheEyeTribe, la cual tiene su dispositivo *lowcost* EyeTribe a un precio de \$100. También podemos encontrarnos con dispositivos que superan los \$45.000, como es el caso del modelo Pro Tx-300 de Tobii. Las diferencias técnicas entre estos dispositivos suelen radicar en la velocidad a la que son capaces de registrar, en los distintos tipos de medidas que realizan, así como en la tecnología empleada en el dispositivo haciendo posible, por ejemplo, que la recuperación de la posición de la mirada sea de manera inmediata, evitando pérdidas de información.

En los últimos años se han realizado varios estudios sobre el rendimiento de los seguidores de la mirada de bajo coste. En 2014, Dalmaijer evaluó la pregunta “¿Es el dispositivo de bajo precio EyeTribe bueno para la investigación?” (Dalmaijer et al., 2014) y concluyó que podía servir para estudios relacionados con mediciones de la pupila y de la fijación de la mirada, pero no para evaluar de manera precisa las medidas de movimientos rápidos del ojo, debido a la baja frecuencia del dispositivo.

La aplicación implementada amplía la realizada por María Reyes en su Trabajo Fin de Master (Reyes Gentil and Varona Martínez, 2017). En este caso, en vez de ser exclusivo para el dispositivo de EyeTribe, se permite también el uso de dispositivos de otras marcas (actualmente sólo está desarrollado para TheEyeTribe y Tobii).

1.2 Objetivos

El objetivo principal en el que se centra este trabajo es el diseño de una aplicación capaz de registrar la actividad ocular de los usuarios durante el proceso de lectura de un texto. Esta aplicación ha de ser capaz de funcionar para diversas marcas y modelos de dispositivos de seguidores de ojos, para poder evaluar y comparar los resultados de las pruebas. Por otro lado, en este trabajo se evalúa la precisión de dos *eye-trackers*, uno de bajo coste y otro de alta gama, durante el seguimiento ocular en tareas de lectura que pueden utilizarse para realizar recomendaciones de diagnóstico de enfermedades como la dislexia y los déficits de atención.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: Introducción.** En esta sección se hará una breve introducción del proyecto, explicando las motivaciones y objetivos.
- **Capítulo 2: Estado del arte.** En esta sección se define lo que es la tecnología de *eye-tracking*, explicando su funcionamiento general y los distintos usos que se le da. Asimismo, se hace una comparativa entre diversos modelos existentes.
- **Capítulo 3: Diseño.** En esta sección se comentarán las decisiones de diseño del proyecto.

- **Capítulo 4: Desarrollo.** En este apartado se describe la funcionalidad de la aplicación, explicando el código del programa. También se muestra el flujo de la aplicación.
- **Capítulo 5: Pruebas y resultados.** En este apartado se describen las pruebas realizadas y se muestran los resultados obtenidos.
- **Capítulo 6: Conclusiones y Trabajo futuro.** Finalmente, se hará una reflexión sobre el trabajo realizado y su posible extensión futura.

2 Estado del arte

En esta sección se describe el estado actual de la tecnología que se va a utilizar para la realización del trabajo.

Primeramente, se definirá lo que es la tecnología de *eye-tracking*, describiendo brevemente su historia, el desarrollo técnico y sus aplicaciones. Posteriormente, se hará una comparativa entre los distintos dispositivos que existen en el mercado, haciendo especial hincapié en los que se han usado en el desarrollo del trabajo. Asimismo, se definirá lo que es la precisión de estos dispositivos y cómo se mide.

2.1 Historia del Eye-Tracking

Eye-tracking (en castellano, “seguimiento ocular”) hace referencia a la técnica usada para grabar y medir los movimientos de los ojos o, lo que es lo mismo, registrar la actividad ocular de un individuo. Puede parecer una técnica bastante novedosa, pero sus orígenes se remontan al siglo XIX, que fue cuando se comenzó a realizar observaciones sobre el comportamiento de los ojos.

En 1879, el oftalmólogo parisino Louis Émile Javal observó que, a diferencia de la suposición tradicional sobre que el movimiento de los ojos a lo largo de un texto era un barrido suave, en realidad se realizaban dos tipos de acciones: fijaciones y movimientos sacádicos (sacadas), es decir, movimientos rápidos de la mirada.

El primer dispositivo (no intrusivo) fue construido por Guy Thomas Buswell, grabando en película los reflejos de haces de luz sobre los ojos. Éste utilizó esta técnica para realizar estudios de lectura y visionado de imágenes. Más tarde (entre los años 50 y 70), el ruso Alfred L. Yarbus, estudió los movimientos sacádicos mientras los individuos del estudio observaban objetos y escenas naturales. Para ello utilizaba la máquina que se observa en la figura 2-1, que, como se puede apreciar, consta de una pieza donde el individuo coloca la cabeza y un par de cámaras que graban los movimientos de los ojos.

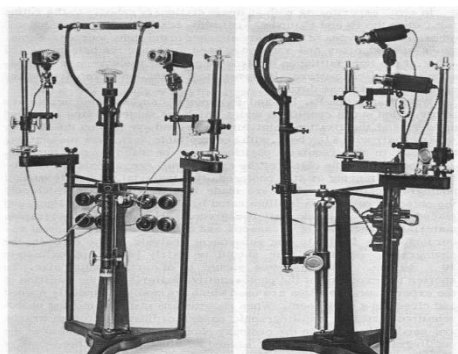


Figura 2-1: Eye tracker de Yarbus

https://upload.wikimedia.org/wikipedia/commons/a/a8/Yarbus_eye_tracker.jpg

En la década de los 70, los dispositivos de seguimiento se volvieron menos intrusivos, proporcionando una mayor precisión y utilizando las imágenes de video del ojo para determinar la localización de la mirada del usuario en la pantalla. Se aumentó el uso de esta técnica para la realización de diversos estudios, especialmente en investigaciones sobre la lectura (Rayner, 1978, 1979).

En la década de los 80, se comenzó a investigar sobre la interacción con los ordenadores con dispositivos de seguimiento ocular. Estos permitían a los investigadores usar la información devuelta por los dispositivos en tiempo real. Este tipo de estudios se enfocaron principalmente en la interacción con gente con discapacidades. A finales de los 90, la agencia de marketing Euro RSCG Worldwide (actualmente conocida como Havas Worldwide) comenzó a usarlo para la medición de las reacciones de los usuarios cuando navegaban por internet (McClatchy, 1998).

En la actualidad, los *eye-trackers* son dispositivos pequeños que proyectan luz infrarroja sobre los ojos y es el reflejo de esta luz lo que se utiliza para determinar la dirección de la mirada. En lo que más se ha mejorado es en la calidad de la imagen y en las técnicas que se utilizan para tomar las mediciones. Se siguen empleando en multitud de áreas, pero destacan los campos de la medicina y los estudios de marketing, tal y como se explicará en el apartado 2.3.

En aplicaciones biomédicas y, dependiendo de las necesidades de cada aplicación, para tener una mayor cantidad de información a la hora de hacer la recolección de datos, los *eye-trackers* se utilizan junto con otros escáneres biométricos como los ECGs, EEG (Kretschmar et al., 2013), etc.

2.2 Eye-Tracker

Los eye-trackers son aparatos que se encargan de registrar y medir las distintas actividades del ojo. En la Tabla 2-1, se describen algunas de estas actividades.

Actividad	Definición
Fijación	Posición estática de los ojos
Movimientos sacádicos	Desplazamientos rápidos de los ojos
Rutas	Combinación de fijaciones y espasmos
Parpadeos	Apertura y cierre de los párpados
Dilatación de la pupila	Variaciones del tamaño de la pupila
Vergencia ocular	Movimiento síncrono (o no) de los ojos
Distancia a la pantalla	Distancia del individuo al monitor

Tabla 2-1: Actividades registradas por los eye trackers

Para ello, utilizan luz infrarroja para provocar un reflejo de la córnea (como ocurre en la mayoría de los dispositivos actuales, aunque hay algunos en los que se ve el reflejo sobre la pupila) y poder así registrar la dirección de la mirada. El motivo de usar luz infrarroja se debe a que estas ondas no pueden ser percibidas por el ojo humano y, además, el espectro de la luz visible hace reflejos en el iris, haciendo que una cámara no pueda detectar con facilidad la dirección de la mirada. Y, como es lógico, los *eye-trackers* tienen una cámara sensible a este tipo de luz para poder grabar el movimiento ocular.

Según su disposición de montaje, hay dos tipos de *eye-trackers*:

Por un lado, están los *remotos* o *estáticos* (los cuales se han utilizado en este trabajo), que se colocan en la parte inferior de un monitor o directamente están acoplados a uno y registran la actividad de los ojos a una cierta distancia del usuario. Estos son los que registran con mayor precisión los datos, pero se ven limitados los movimientos del individuo, puesto que el *eye-tracker* está colocado en una posición fija y, al mover la cabeza, se pueden ver afectados los resultados.

Por otro lado, están los *móviles*, que son dispositivos que se colocan en la cabeza y se utilizan más para los estudios de mercado, de atención, etc. A diferencia de los otros, estos permiten una mayor libertad de movimiento al individuo, pero, al ser más intrusivos, pueden ser más incómodos.

Algunos de los fabricantes de estos dispositivos son: SMI, EyeTech, Mirametrix, ASL, GazePoint, SR Research, SensoMotoric Instruments, Tobii y The Eye Tribe. Y en el caso de este trabajo, se van a usar dispositivos de los dos últimos fabricantes.

Como es lógico hay diferencias entre los distintos *eye-trackers* que se pueden encontrar en el mercado y, por ello, a continuación, se indican las especificaciones de algunos modelos:

- **Tobii Pro Glasses 2:** El *eye-tracker* móvil de Tobii, que tiene cuatro cámaras para registrar los movimientos de los ojos, más otra para grabar lo que está viendo el usuario en resolución Full HD. Captura los datos a 50 o a 100 Hz. Esta segunda generación es bastante más ligera que su antecesora, pesando sólo 45 gramos.



Figura 2-2: Tobii Pro Glasses 2

http://www.tobiipro.com/imagevault/publishedmedia/jn0knq8unotexk30z8b5/TobiiPro_Glasses_2_Eye_Tracker_side_3_1.jpg

- **Tobii X2-30/X2-60:** Se trata de un *eye tracker* remoto. Ofrece una precisión de hasta 0'4°. Existen dos versiones del modelo X2; la diferencia entre ambos es la frecuencia de captura de los datos: 30 o 60 Hz. Entre los datos que recoge se encuentran: la posición de los ojos respecto a la cámara, la posición de la mirada en pantalla y el diámetro de la pupila.



Figura 2-3: Tobii X2-30

http://www.tobiipro.com/imagevault/publishedmedia/99zxc4h39mu8sfvbu44/Tobii_Pro_X2_Eye_Tracker_3_1.jpg

- **TheEyeTribe Tracker:** Es un *eye-tracker* remoto de bajo coste (\$100) de la compañía EyeTribe. Recoge información como la posición en pantalla de la visión, el diámetro de la pupila y la posición de los ojos respecto a la cámara. Tiene una precisión entre 0'5° y 1° y captura los datos a 30 o 60 Hz. Este es uno de los dispositivos que se usarán en el desarrollo del trabajo. Utiliza rayos infrarrojos para determinar la posición de la mirada, a partir del reflejo sobre los ojos.



Figura 2-4: The Eye Tribe tracker

https://s3.eu-central-1.amazonaws.com/theeyetribe.com/theeyetribe.com/lh5.googleusercontent.com/demo-mode_minimuzwd.jpg

- **Tobii Pro Tx-300:** Es un dispositivo todo en uno, es decir, viene integrado en una pantalla de 23". Recoge información como la posición en pantalla de la visión, el diámetro de la pupila y la posición de los ojos respecto a la cámara. Tolera los movimientos de la cabeza y tiene recuperación inmediata de la mirada al parpadear. Tiene una precisión de 0'14° y puede capturar datos a 60, 120 o 300 Hz. La técnica de seguimiento es binocular y del negro de la pupila.



Figura 2-5: Tobii Pro Tx300

http://www.tobiipro.com/imagevault/publishedmedia/nfk0u0rxgr964ymhk7iw/TobiiPro_TX300_Eye_Tracker_3_1.jpg

En concreto, en la realización de este trabajo se han usado el Tobii x2-30, el Tobii Pro Tx300 y el EyeTribe, aunque sólo han sido usados los dos últimos para la generación de resultados, mientras que el Tobii x2-30 fue usado durante el desarrollo de la aplicación, y para la realización de las pruebas. En la tabla 2-2, se comparan distintos aspectos que caracterizan a estos dispositivos.

	Tobii x2-30	Tobii Tx-300	EyeTribe
Precisión	0.4°	0.14°	0.5° y 1°
Tolerante a movimientos	Baja	Alta	Baja
Recuperación tras parpadeo	Inmediata	Inmediata	Rápida
Frecuencia de muestreo (fps)	40	60, 120 o 300	30 o 60
Incluye monitor	No	Si	No
Tamaño de monitor recomendado	Hasta 25''	Sistema integrado de 23''	Hasta 27''
Software disponible	- SDK - Tobii Studio (no incluido en el precio)	- SDK - Tobii Studio (no incluido en el precio)	- SDK - Herramienta de calibración gratuita.
Técnica de rastreo	Rastreo del reflejo de rayos infrarrojos sobre las pupilas.	Rastreo del reflejo de rayos infrarrojos sobre las pupilas.	Rastreo del reflejo de rayos infrarrojos sobre las pupilas.
Información registrada	- Momento del registro. - Posición de la mirada de los ojos. - Tamaño de la pupila.	- Momento del registro. - Posición de la mirada de los ojos. - Tamaño de la pupila.	- Momento del registro. - Posición de la mirada de los ojos. - Tamaño de la pupila. - Posición del centro de la pupila. - Comprobación de si la mirada está fijada.
Coste	\$1.100 al mes para alquilar. \$20.000 para comprar	\$4.900 al mes para alquilar. \$45.000 para comprar	\$99

Tabla 2-2: Comparativa de los seguidores de mirada usados

En el caso de los dispositivos de Tobii el precio no está especificado en la página del fabricante, pero se puede encontrar en diversas páginas no oficiales para el modelo x2-30 y, para el Tx-300, hay un documento oficial de Tobii (Tobii, 2012). En ambos casos, son documentos de hace al menos 2 años y, por consiguiente, el precio puede haber variado.

El *software* mencionado en la tabla 2-2, Tobii Studio, es el programa que tiene la empresa Tobii para realizar la calibración, registro de actividad, realización de pruebas de reacción de estímulos y, para acceder a estas prestaciones, es necesario tener una licencia que cuesta \$4.700 para la edición básica, \$8.900 para la edición profesional y \$12.900 para la versión empresarial, además de un coste adicional al año si se quiere tener las últimas actualizaciones del *software*, que oscila entre los \$1.800 y \$5.000.

2.3 Campos en los que se emplea la tecnología eye-tracking

Esta técnica se utiliza en multitud de ámbitos, desde la medicina hasta los videojuegos. Para empezar, uno de los principales campos en el que se utiliza es en los estudios de mercado, viendo las cosas en las que se fijan los consumidores mediante el uso de *heatmaps*. Ya sea monitorizando lo que está viendo en una búsqueda por internet (véase la Figura 2-7), o lo que está mirando cuando va una persona al mercado (véase la Figura 2-6), el uso de esta tecnología ayuda a caracterizar el comportamiento del consumidor y saber en qué cosas se fija. Para este tipo de estudios, se usan distintos tipos de dispositivos. En la tecnología actual, para el caso del visionado de contenido en internet, se utilizan mayormente los *eye-trackers* remotos, mientras que en los estudios de mercado in situ, al estar en movimiento, se requiere del uso de dispositivos móviles.

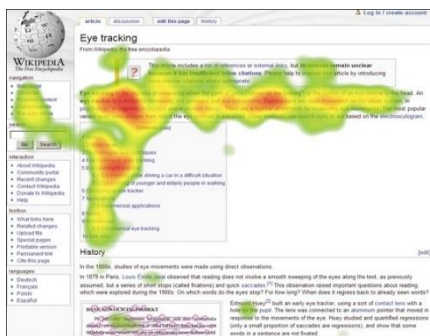


Figura 2-7: Heatmap página web
<https://www.godaddy.com/garage/wp-content/uploads/2017/03/heat-map-examples-eye-tracking.jpg>



Figura 2-6: Heatmap haciendo la compra
<https://canadasbeststorefixturesblog.files.wordpress.com/2013/07/heatmap.png>

En el campo de la medicina, la tecnología de seguimiento ocular se ha utilizado para el diagnóstico de enfermedades neurológicas, como, por ejemplo, déficit de atención e hiperactividad, esquizofrenia. (Holzman et al., 1973; Jacobsen et al., 2017), alzhéimer (Pereira et al., 2014), etc.

En la enseñanza se ha utilizado para hacer estudios de atención a la hora de leer, ya sea viendo las fijaciones que se hacía sobre las palabras (véase la Figura 2-8), como para analizar las veces que se retrocedía para releer una parte del texto, y así ver el efecto que tenía sobre el rendimiento de los estudiantes (Zawoyski et al., 2015). También se ha usado para ver en qué se centran los estudiantes durante una presentación, habiéndose observado que, por lo general, prestaban más atención a las imágenes más relevantes (Slykhuus et al., 2005).

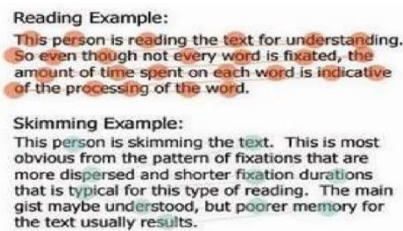


Figura 2-8: Eye tracking para detección de patrones de lectura
https://instagloss.com/static/reading_skimming.jpg

Otro sector que ha incorporado esta tecnología es el de la experiencia del usuario, permitiendo la interacción de las personas con las máquinas, usando la mirada como si fuera, por ejemplo, el ratón del ordenador. También se puede usar en algunos videojuegos.; por ejemplo, Tobii ya cuenta con un par de dispositivos especializados para jugar, el *Tobii EyeX* y el *Tobii EyeTracker 4C*; y tiene más de 50 títulos compatibles con esta tecnología.

2.4 Precisión

La precisión de los dispositivos de *eye-tracking* no corresponde a una medida única, pues hay varios factores a tener en cuenta. La medida que se utiliza para expresar la precisión tradicionalmente son los grados (siendo un grado un error de 12 milímetros aproximadamente sobre un monitor que está a una distancia de 65 cm (Tobii TX300 Eye Tracker)).

Para empezar, tenemos la técnica que mide cuán desfasadas están las mediciones realizadas. También está la precisión de la mirada, es decir, la desviación que hay entre la posición en la que el dispositivo detecta la mirada en un instante y en los siguientes.

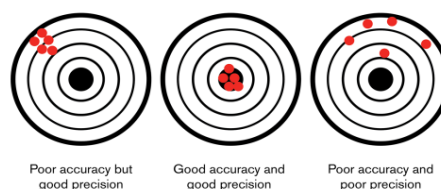


Figura 2-9: Precisión de la mirada
http://www.tobiipro.com/imagevault/publishedmedia/pu6jv5uf9wpt7v3og1k4/Accuracy_Precision.png

Los test de precisión y exactitud que realizó el fabricante de *eye-trackers* Tobii (Tobii® Technology Accuracy and precision test method for remote eye trackers, 2011) especificaron que la precisión podía ser calculada con la siguiente fórmula:

$$Precisión = \sqrt{\frac{1}{n} \sum_{i=1}^n \theta_i^2}$$

Donde Θ_i es la diferencia entre un ángulo de visión en un instante y el siguiente, tal y como se puede apreciar en la siguiente imagen.

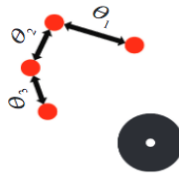


Figura 2-10: Muestreo precisión de la mirada <http://www.tobii.com/siteassets/tobii-pro/learn-and-support/use/what-affects-the-performance-of-an-eye-tracker/tobii-test-specifications-accuracy-and-precision-test-method.pdf?v=2.1.1>

La precisión, para el caso de los *eye-trackers* de Tobii, se calcula de dos maneras, por ojo o como la media de los dos ojos. Para que esta sea óptima, se deben tener en cuenta factores externos al dispositivo que pueden repercutir en los resultados. Por ejemplo, la iluminación y la temperatura del lugar en el que se use el dispositivo, problemas propios de los ojos (estrabismo, uso de gafas o lentillas, haberse hecho algún tipo de intervención quirúrgica, etc.), la posición de la cabeza y una buena calibración antes de usarlo. También es conveniente eliminar cualquier tipo de distracción que pueda surgir durante el estudio, por ejemplo, las notificaciones de aplicaciones (como las del antivirus u otras aplicaciones que envíen mensajes).

Como es lógico, la calidad del dispositivo también influye. Por ejemplo, la resolución de la imagen y la frecuencia de muestreo podrán afectar a los datos recogidos. Y esto es algo en lo que pueden diferir bastante los dispositivos de distintos rangos de precios. En concreto, el dispositivo de TheEyeTribe (de bajo coste), tiene una frecuencia de muestreo máxima de 60Hz, bastante más bajo que el Tobii Pro Tx-300 (de coste más elevado), cuya frecuencia tope es de 300Hz, tal y como se indicó en el apartado 2-2.

En los últimos años se han realizado varios estudios sobre el rendimiento de los seguidores de la mirada de bajo coste, por si solos o frente a otros aparatos más caros. En 2014, Dalmaijer evaluó la pregunta “¿Es el dispositivo de bajo precio EyeTribe bueno para la investigación?” (Dalmaijer et al., 2014) y concluyó que podía servir para estudios relacionados con mediciones de la pupila y de la fijación de la mirada, pero no para evaluar de manera precisa las medidas de movimientos rápidos del ojo, debido a la baja frecuencia del dispositivo.

Asimismo, se hizo un estudio sobre qué *eye-trackers* eran adecuados para la investigación, haciendo un análisis sobre el rendimiento de distintos dispositivos de múltiples precios (Funke et al., 2016) y lo que experimentaron fue que los dispositivos mantienen una precisión y exactitud aceptable, pero tuvieron una alta tasa de pérdida de datos.

3 Diseño

En esta sección se describe el diseño de la aplicación realizada en este trabajo. Tal y como ya se ha mencionado en el apartado 1.2, el objetivo principal es el diseño de una aplicación capaz de registrar la actividad ocular de los usuarios durante el proceso de lectura de un texto, funcionando para dispositivos de distintas marcas y modelos de seguidores de ojos, para poder comparar los resultados de las pruebas de lectura orientadas al diagnóstico de dislexia o déficits de atención.

3.1 Dispositivos

Tal y como ya se ha mencionado en la sección 2-2, hay una gran variedad de marcas y aparatos, pero para este trabajo, se han utilizado tres dispositivos: el dispositivo de EyeTribe y los modelos x2-30 y Tx-300 de Tobii, aunque para simplificar las pruebas con los voluntarios no se haya usado el x2-30 (pero si se ha usado durante el desarrollo de la aplicación), priorizando la mayor resolución del Tx-300. Y, por ello de aquí en adelante ya sólo se hablará de los dos dispositivos usados en las pruebas.

Los motivos del uso de estos dispositivos han sido la diferencia de rangos de precios y prestaciones, para ver si la diferencia de precio realmente merece la pena para la realización de pruebas y, por otro lado, estaba la disponibilidad de los mismos en el laboratorio del Grupo de Neurocomputación Biológica y en el Laboratorio de Visión Humana de la UAM.

En el Anexo A se incluye una guía de instalación de los dispositivos.

3.1.1 Colocación de los dispositivos

En primer lugar, el dispositivo de Tobii Tx-300 está integrado en un monitor, lo que facilita mucho el manejo del dispositivo ya que sólo es necesario realizar el proceso de calibración. Este dispositivo es voluminoso (ver figura 2-5), algo que puede restringir su uso en algunos contextos. Para ponerlo en funcionamiento, después de hacer el proceso de configuración explicado en el Anexo A, lo único que hay que hacer es conectar el cable ethernet del dispositivo en el que se quieren registrar los resultados. Tal y como se puede apreciar en la figura 3-1, se puede desacoplar y usar de diversas maneras, aunque para el presente caso, se usará de manera acoplada al monitor.

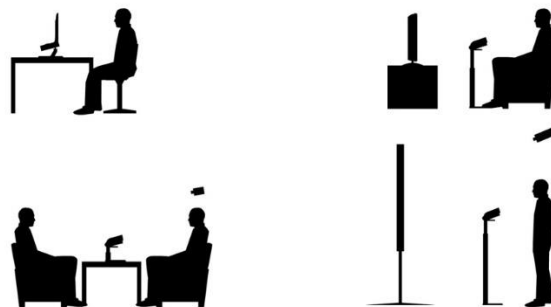


Figura 3-1: Colocación dispositivo Tobii

https://www.tobiipro.com/imagevault/publishedmedia/s48hj6kt1j174almtf4s/TX300_setup_2_1.jpg

Por otro lado, el dispositivo de EyeTribe, de dimensiones reducidas (ver figura 2-4), hay que colocarlo sobre el trípode que viene en la caja y se conecta al ordenador mediante una conexión de *USB 3.0* (obligatoriamente, ya que no funciona si se conecta a un puerto de una versión inferior). El dispositivo se debe colocar de manera que quede lo más centrado posible

respecto a la pantalla y a una distancia que, recomendablemente, no sea superior a 60 cm de la persona a la que se quiere grabar el registro de actividad.

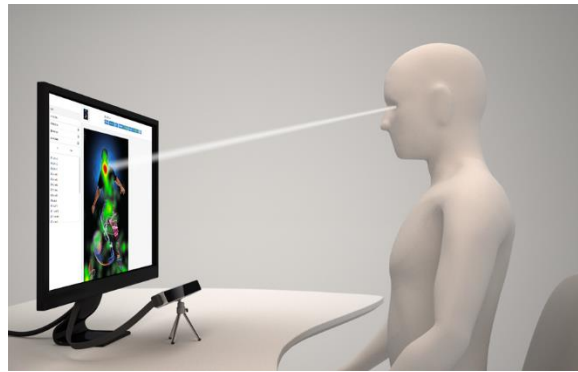


Figura 3-2: Colocación dispositivo EyeTribe <https://s3.eu-central-1.amazonaws.com/theeyetribe.com/theeyetribe.com/dev/general/index.html>

3.1.2 Calibración

Para el dispositivo de Tobii, en principio se debería haber realizado el proceso de calibración con la aplicación de Tobii Studio. No obstante, se ha utilizado la aplicación de ejemplo de calibración incluida en el *development tools*, puesto que era más fácil de integrar en la aplicación desarrollada en este trabajo, ya que era una aplicación que se encargaba específicamente de esta tarea e interfería menos con el resto de la prueba. Además, el proceso de calibración era muy rápido (menos de 10 segundos) y solía tener buenos resultados en la mayor parte de los casos. Los resultados se muestran tal y como se puede ver en la figura 3-3. Para saber si el proceso se completó satisfactoriamente, hay que fijarse en las líneas que hay en los puntos. Si las líneas son pequeñas y apenas salen de los círculos, significa que la calibración es válida para la realización de las pruebas.

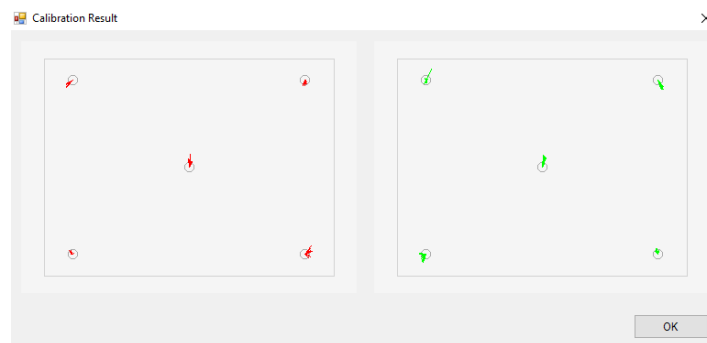


Figura 3-3: Calibración Tobii

Por otro lado, para el dispositivo de EyeTribe se ha utilizado el programa de calibración que venía incluido en el programa de instalación del SDK. En este caso, la calibración dura un poco más, pero también depende del número de puntos que se pongan para realizar la calibración ya que permite la selección de 5, 10 o 16 puntos, haciendo que la calibración fuera más precisa cuanto mayor número de puntos se usaran. A diferencia del de Tobii, este protocolo de calibración proporciona un resultado de la calidad a modo de estrellas, de 0 a 5 (siendo 0 repetir la calibración y 5 una calibración perfecta). Además, este dispositivo permite que, cuando se ha terminado la calibración, se iluminen los puntos de la pantalla

según se pasa la mirada por encima, lo cual ayuda a cerciorarse de la calidad del resultado de la calibración.



Figura 3-4: Calibración EyeTribe

3.1.3 Datos proporcionados

Para la recogida de los datos se ha usado el SDK de .Net, el cual devuelve toda la información en estructuras de datos.

Por lo general ambos dispositivos recogían los mismos datos, aunque algunos expresados de diferentes maneras. Los datos que tenían en común son los mostrados en la tabla 3-1.

	EyeTribe	Tobii
Time Stamp	Tiempo en milisegundos	Tiempo en microsegundos
Time Stamp String	Cadena de texto con el momento del registro	N/A
Información por cada ojo	<ul style="list-style-type: none"> - Coordenadas (en píxeles) de la mirada - Coordenadas suavizadas (en píxeles) de la mirada - Coordenadas en píxeles de la pupila - Tamaño de la pupila 	<ul style="list-style-type: none"> - Coordenadas (en tanto por uno de la pantalla) de la mirada - Coordenadas (en tanto por uno de la pantalla) de la pupila - Tamaño de la pupila
State	Código del estado del aparato (registrando, perdido, presente)	N/A
Fix	Booleano para saber si la mirada está fija	N/A

Tabla 3-1: Datos proporcionados por los SDK

3.2 Usuarios

La aplicación desarrollada está pensada para que la pueda utilizar cualquier persona que tenga al menos el nivel de lectura de un niño de primaria y conocimiento de la prueba que se está realizando, que se puede obtener en el momento de la prueba, puesto que se incluye una descripción de esta, así como del funcionamiento de la aplicación.

3.3 Aplicación

La aplicación ha sido desarrollada de manera íntegra en la plataforma .Net, mediante el uso del IDE de Visual Studio (<https://www.visualstudio.com/es/>) y el lenguaje de programación C#. En el Anexo B se puede ver un diagrama de clases de la aplicación realizada.

3.3.1 Universalidad de uso

Se ha desarrollado la aplicación intentando que todo fuera lo más genérico posible, para permitir ampliaciones, adaptaciones o mejoras en el futuro.

En cuanto al diseño de la aplicación, se ha hecho para que se adapte a cualquier tamaño de monitor (recordando que es una aplicación de escritorio), por lo que las limitaciones de la pantalla serán las que tengan los dispositivos.

Por otra parte, para la implementación del código de los *eye-trackers*, se ha creado una clase abstracta con los métodos principales necesarios para que se puedan utilizar con la aplicación:

- *Calibrate*: para realizar la calibración del dispositivo.
- *StartRecording*: para que el dispositivo empiece a grabar y mandar los datos a la aplicación.
- *CloseConnection*: para finalizar el registro de la actividad.
- *OnGazeUpdate*: este depende de las librerías de los dispositivos, ya que reciben las estructuras de datos con el formato pertinente del dispositivo en cuestión.

Con ello se consigue que, si se quiere utilizar un nuevo dispositivo, sólo se tendrían que implementar estas funciones y hacer los ajustes en los menús de la aplicación para que aparezca entre las opciones de los dispositivos a usar.

Asimismo, en la aplicación, todas las rutas que se usan son relativas al iniciar la aplicación (después, las rutas seleccionadas a lo largo del proceso de configuración se volverán absolutas), por lo que funcionará el código en cualquier ordenador.

3.3.2 Distribución del código

El código está modulado en carpetas en función de la actividad para la que estén destinados los ficheros:

- *CalculoPrecision*: tiene todos los objetos usados para el cálculo de la precisión.
- *EyeTrackers*: tiene la implementación de los registros de la actividad ocular. Si se quisiera implementar para un nuevo dispositivo, se colocaría aquí y se haría que extendiera la clase abstracta EyeTracker.
- *GeneracionResultados*: contiene el código usado para la generación de los ficheros de salida: las imágenes, videos y el fichero csv con los cálculos de la precisión.
- *InterfazGrafica*: incluye el diseño de los menús de la aplicación, así como la lógica del programa.
- *Librerias*: contiene las clases creadas para la interacción con aplicaciones externas (Gnuplot, ffmpeg, el programa de calibración de Tobii y la grabadora de audio de Windows).

3.4 Ficheros de lectura

Los ficheros de lectura que se utilizan en las pruebas se encuentran en el directorio principal del proyecto, en una carpeta llamada “Reading files”, aunque se permite la selección de cualquier fichero que esté en el equipo, siempre y cuando sea un archivo *png*. Además, se requiere que haya un *csv* en la misma carpeta en la que vengan los vértices de las líneas calculadas en tanto por uno de la pantalla, para poder realizar el cálculo de la precisión. En el apartado 5.2.1, se describen más en detalle estos archivos.

3.5 Grabación del audio

Para realizar la grabación del audio se ha usado el paquete NAudio, el cual se puede descargar con NuGet, que proporciona todas las herramientas necesarias para grabar el audio.

3.6 Calibración

El proceso de calibración se realiza a través de las aplicaciones dadas por los fabricantes, tal y como se explica en los Anexos C (calibración del dispositivo de TheEyeTribe) y D (calibración del dispositivo de Tobii).

3.7 Resultados

Los resultados se guardan en la carpeta “Results”, la cual está en el directorio principal del proyecto. En esta carpeta se crean las subcarpetas para cada una de las pruebas. El nombre de estas será el código SHA generado de la combinación del nombre, apellidos y fecha del registro de la prueba, para que sea único y añadir privacidad a los resultados de los sujetos. En cada una de las carpetas se generan los siguientes archivos:

- “Left_full.png” y “Right_full.png”: gráficos en los que aparece la actividad completa de cada uno de los ojos registrada por el dispositivo.
- “Left.png” y “Right.png”: son la versión reducida de los anteriores, la única diferencia es que se le ha pasado un filtro a los datos para que sólo estén los valores no nulos y comprendidos en las dimensiones de la pantalla, para evitar valores inválidos que dificulten la visión de la actividad.
- “Left_color.png” y “Right_color.png”: muestran los mismos datos que las versiones reducidas, pero con una escala de color que codifica el paso del tiempo en los trazos del seguimiento ocular.
- “Left_pupil.png” y “Right_pupil.png”: muestran la variación del tamaño de la pupila a lo largo del tiempo de la prueba.
- “Left_movie.mpeg” y “Right_movie.mpeg”: son los videos que muestran el avance del patrón de lectura con el audio de la grabación realizada durante el proceso de registro.
- “Precision.csv”: fichero Excel que contiene información que permite ver de manera rápida la calidad de una prueba. Contiene el número de retrocesos de la mirada, el porcentaje de posiciones coherentes detectadas, así como la distancia media de fallo (en píxeles). En el apartado 4.2.2.2 se explica cuál ha sido el procedimiento para calcular estos valores.

Asimismo, se genera una línea en un fichero *txt* para llevar el registro de las pruebas, guardando la información necesaria de estas: nombre completo del usuario, directorio en el que se han guardado los resultados, dimensiones de la pantalla (en píxeles), marca del

dispositivo usado, frecuencia de muestreo empleada en el registro, fecha de inicio de la prueba, su duración y el *path* del fichero de lectura empleado en la prueba.

4 Desarrollo

En esta sección se indica cómo se ha desarrollado la aplicación, comentando las tecnologías escogidas para su implementación.

4.1 Herramientas usadas

El SDK de EyeTribe está disponible para Java, C++, Python y .Net; mientras que el de Tobii está para .Net, Python, Matlab y C.

Para el desarrollo de la aplicación en este trabajo se ha empleado el lenguaje de .Net, C#. Por ello, para el desarrollo se ha optado por la utilización del IDE de Visual Studio, el cual se puede descargar desde <https://www.visualstudio.com/es/downloads/>.

Para que el programa funcione, ha de estar instalado el SDK de EyeTribe, que se puede encontrar en <https://github.com/EyeTribe/sdk-installers>.

También es necesario, para la generación de los gráficos y películas del seguimiento ocular, disponer de Gnuplot (<http://www.gnuplot.info/>) y Ffmpeg (<https://ffmpeg.org/download.html>), aunque no es necesario descargarlos e instalarlos, puesto que ya han sido incluidos dentro de la carpeta “bin” del proyecto con todas sus dependencias.

El otro programa externo que requiere la aplicación es el que viene con el *SDK analytics tool* de Tobii, del cual se incluye una copia con el código, ya que no se puede descargar ahora que han sacado un nuevo SDK.

Una vez tenemos estos programas, se puede proceder a la instalación y configuración.

1. Se instala Visual Studio para .Net para la realización de proyectos de Windows Forms.
2. Se crea un proyecto: *Archivo > Nuevo > Proyecto*.
3. Se elige en *Plantillas > Otros lenguajes > Visual C#* la opción Aplicación de Windows Forms.
4. Una vez creado el proyecto se hace click con el botón derecho sobre Referencias y se clicla la opción “Administrar paquetes NuGet...”
5. Se instalan los siguientes paquetes: *CircularProgressBar*, *EyeTribe.ClientSdk*, *NAudio* y *Tobii.Researchx86*.

Con ello ya se tiene todo lo necesario para editar y usar el programa.

4.2 Aplicación

La aplicación tiene cuatro partes importantes: registro del usuario, calibración, registro de patrones y generación de los resultados.

El registro del usuario se realiza a través de la interfaz gráfica. En el caso de la calibración, se han empleado las aplicaciones proporcionadas por los fabricantes, y la aplicación creada realiza las llamadas pertinentes a estos programas. Por ello, esta subsección se centra en el

resto de la aplicación (que es la funcionalidad que se ha desarrollado), y se hará referencia a las imágenes del Anexo E. Y será en la sección 4.3, en la que se explicará el funcionamiento completo de la aplicación.

El código de la aplicación, junto con los resultados, puede ser accesible a través del siguiente enlace: <https://mega.nz/#F!8JMEkQ5Q!Bb3jcmP8MrAbfs5PhIeDcw>

4.2.1 Registro de patrones

Tal y como se menciona en el apartado 3.3.1, el código del registro de patrones se encuentra en una clase abstracta en la que hay dos funciones principales definidas. Estas son la inicialización de los dispositivos y ficheros, así como el proceso de almacenamiento de los datos en los ficheros establecidos.

InitRegistration es la función que se encarga de inicializar todos los dispositivos de grabación, tanto el de audio como el *software* del *eye-tracker*, y de comenzar la grabación. Al realizar la llamada del método “*StartRecording()*”, que se puede ver en la figura 0-10, se inicia el registro de la actividad ocular antes que la de la grabación del audio. Esto se debe a que, durante el desarrollo, se experimentaron diversos problemas relacionados con los tiempos de registro y los de las grabaciones de audio cuando se inicializaba antes el grabador de audio. Esto se solventó cambiando el orden de las llamadas a las funciones. Asimismo, en esta función, también se crean los ficheros en los que se van a almacenar los resultados que serán usados por la función de generación de resultados.

Por otro lado, está la función *RegisterOcularActivity*, la cual se encarga de guardar, en los ficheros de resultados, la actividad ocular. Esta función se ha modificado durante el desarrollo por diversos motivos. Inicialmente, la aplicación registraba la actividad durante un periodo definido durante el proceso de configuración de la prueba. Para ello, se usaba el temporizador que proporciona *Windows Forms* para ejecutar la función, ya que permitía indicar la frecuencia a la que se deseaba que se ejecutara y, pasando por parámetros el momento de tiempo en el que se ejecutó la función, se podía llevar un registro de la temporalidad. El problema llegó cuando se observó que el programa no finalizaba en el tiempo que se le decía y también se comprobó que, cuanto mayor era la frecuencia de registro, peor era el comportamiento.

La siguiente aproximación consistió en la finalización del registro en el momento en que el usuario terminara la lectura. Para ello, se intentó utilizar el reloj interno del ordenador, haciendo llamadas a funciones nativas de C#. No obstante, dio algunos problemas y, al final, se decidió utilizar el *timespan* que proporcionan los dispositivos, incluyendo una versión de los tiempos en microsegundos y otra en milisegundos, ya que cada dispositivo utilizaba unas unidades distintas. No obstante, el tenerlo en dos unidades se aprovechó haciendo que se cogiera el valor en microsegundos para comprobar si el registro era el mismo que el anterior, y así evitar duplicados; y el valor en milisegundos para guardar la marca de tiempo en el fichero, quedando esta función tal y como se ve en la figura 0-11. También comentar que, para que los datos entraran a la versión reducida, debían cumplir que estuvieran los valores de los ojos en una posición lógica, es decir, entre los valores de las dimensiones de la pantalla, y tampoco aceptando valores nulos.

Otro problema que surgió fue cómo parar el proceso de registro. Para empezar, como se llamaba a esta función realizando un bucle infinito, el cual sólo finaliza cuando se pulsa una tecla, se tuvo que crear un hilo aparte para que no se congelara la aplicación. Esto supuso

otro problema. Al final del registro se debía de cerrar la ventana, pero, al estar ejecutándose en un hilo aparte, fue necesario recurrir al *SynchronizationContext* para poder hacer la llamada desde el hilo principal, tal y como se ve en la figura 0-12. El único inconveniente, si se puede considerar como tal, es que *OnKeyPress* responde lentamente para todas las teclas salvo para el ESC, cuya respuesta es inmediata.

En lo relativo a las funciones de actualización de los datos obtenidos por los dispositivos, hay poca diferencia entre ellas. Lo único a destacar es que el dispositivo de EyeTribe proporciona más datos que los dispositivos de Tobii, tal y como se puede ver en las figuras 0-13 y 0-14 y se detalla en la tabla 3-1, aunque a una frecuencia bastante inferior, ya que la frecuencia a la que trabaja el dispositivo de EyeTribe es 10 veces menor que la del Tobii Tx-300.

Para solventar el déficit de información, se decidió generar los datos que el SDK no traía pero se podían inferir a partir de otros, como era el caso de si la mirada está fija, comprobando si habían variado las posiciones de un instante al siguiente. Y, para los valores no conocidos como el estado y las posiciones de las pupilas, se dejaron con un valor por defecto ya que, aunque para este estudio estos datos no han sido usados, si en el futuro se quiere ampliar la aplicación y usar esa información, o se incorpora otro dispositivo que sí que posea esos datos, estén ya disponibles.

4.2.2 Generación de Resultados

Los resultados de la aplicación se describen en dos apartados distintos. Por un lado, la generación de los ficheros de salida, gráficos y vídeos, y, por otro, el cálculo de la precisión del dispositivo en la prueba realizada a partir de los datos generados.

4.2.2.1 Generación de los resultados multimedia

Para la generación de los resultados se ha usado la herramienta de generación de gráficos *GnuPlot*. La cual ha sido integrada en la aplicación y se pueden ejecutar los comandos de la misma manera que se hace a través de la aplicación propia. Los datos que se utilizan para elaborar dichas gráficas son los procedentes de los ficheros generados en la funcionalidad del registro de patrones. Puesto que el código de las gráficas es idéntico para ambos ojos, a excepción de los directorios y columnas usadas, sólo se mostrará el de uno de ellos.

Para que apareciera la imagen que se usó en la prueba de fondo de pantalla como fondo de las gráficas, se utilizó el código de la figura 0-15 en el que se implanta la imagen escalada a la dimensión del monitor en el que se hizo la prueba, que es el tamaño de los ejes (donde *propX* y *propY* son las proporciones del tamaño de la imagen respecto al del monitor).

Para empezar, se genera la imagen que muestra la actividad completa registrada ("*full_reading_pattern.txt*") por el dispositivo con el código de la figura 0-16, generando los ficheros "*Left_full.png*" y "*Right_full.png*".

Después se generan los ficheros equivalentes, pero para la versión reducida ("*reduced_reading_pattern.txt*"). Esta versión es la usada para el resto de gráficas que se comentarán posteriormente.

Seguidamente, en el código de la figura 0-17 se generan las versiones equivalentes a las anteriores, pero en las que aparece con la temporalidad marcada con códigos de colores, según se muestra en la escala que aparece al lado de los gráficos.

A continuación, se sacan los gráficos que muestran la evolución del tamaño de la pupila a lo largo de la prueba (figura 0-18).

Por último, en el código expuesto en la figura 0-19, se generan los gráficos que permiten la creación de los videos, junto con el archivo de audio que se genera en la recolección de los datos. Puesto que es un proceso bastante costoso, se ha limitado el número de *frames* por segundo a 30, que es el mínimo de las frecuencias registradas y más que suficiente para apreciar el movimiento. Esto hace que se generen un total de $(30 * \text{tiempo (s)})$ fotogramas. Con estos fotogramas y el fichero de audio se genera el video.

4.2.2.2 Cálculo de la precisión

Para calcular la precisión se generó, para cada una de las imágenes, un fichero csv en el que se guardaron las posiciones de las esquinas de las líneas del texto de lectura, para poder tener una idea de la ubicación de estas en la imagen. Las posiciones se registran en tanto por uno de la dimensión de la imagen, para poder usar la aplicación con cualquier resolución de pantalla y que los resultados sean coherentes.

Para saber la distancia de un punto a una línea, lo que se hace es comprobar primero si el punto está en el rectángulo que forman los puntos de las esquinas de la línea. Si está dentro, la distancia es 0. Si no, la distancia es calculada de la siguiente manera: se busca la menor de las distancias entre las líneas verticales para sacar la distancia en el eje x y la menor de las distancias entre las líneas horizontales para sacar la distancia en el eje y. Con esto, se calcula la distancia euclídea y se toma como la distancia del punto a la línea. Tal y como se aprecia en la figura 0-20, se calcula cuál es la línea más cercana y la distancia a esta para cada uno de los puntos registrados.

Puesto que en la prueba se han de leer todas las líneas del texto, se puede asegurar que siempre se ha de empezar en la primera línea y terminar por la última. Esto permite saber si hubo algún tipo de desplazamiento vertical que afecte a la precisión tal y como se puede ver en la figura 0-21.

Después, tal y como se muestra en la figura 0-22, se limpian las observaciones previas a la colocación de la mirada sobre la primera línea, para no ensuciar el resultado. Si hubo desplazamiento hacia arriba, la primera línea no tendrá muestras casi seguro, así que se toma por primera observación válida la primera de todas.

Una vez hecho todo esto, se calcula el número de puntos coherentes tal y como se expone en el Anexo F.

Tal y como se muestra en la figura 0-23, el método del cálculo de la precisión devuelve un objeto con todos los datos que se almacenan en el fichero “precisión.csv”. Este código se ejecuta para las muestras de los dos ojos y también se calcula la media, para calcular una precisión media de la prueba que sea válida para la comparación de los resultados.

4.3 Interfaz Gráfica

La aplicación requiere que haya al menos un micrófono conectado al ordenador en el que se vaya a ejecutar el programa. En caso contrario, saltará el mensaje de error que se muestra en la figura 4-1, y no llegará a abrirse.

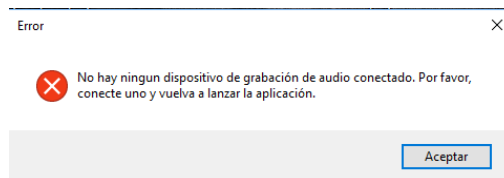


Figura 4-1: Error "No hay micrófono conectado"

Si hay un dispositivo conectado que permita la grabación de audio, aparecerá la ventana principal del programa mostrada en la figura 4-2.

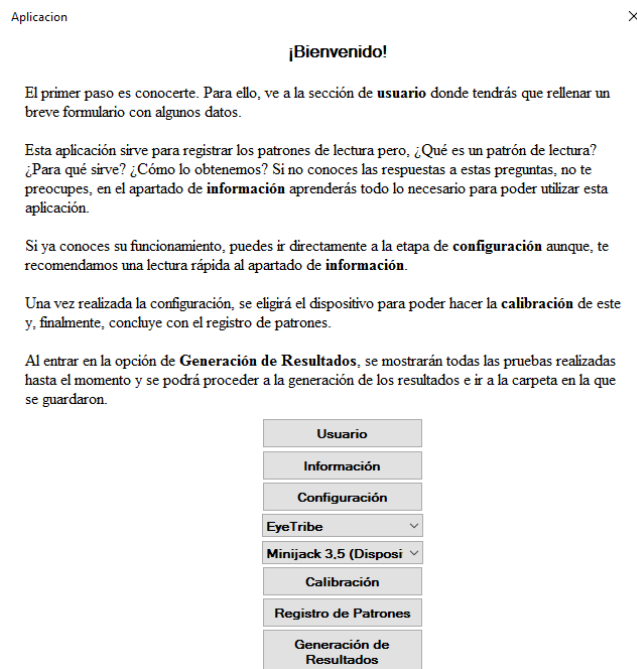


Figura 4-2: Menú principal

Aquí se puede ver que hay dos secciones. La primera es un texto en el que se describe el procedimiento que ha de seguirse para realizar correctamente la prueba y luego están las opciones de menú, que son las tareas que han de realizarse para efectuar la prueba de los dispositivos.

La única dependencia para acceder a uno de estos pasos se da en el “Registro de Patrones” ya que, para poder realizarlo, es obligatorio haber realizado el registro del usuario, la configuración de la prueba y la calibración del dispositivo. En caso de que falte por realizar alguno de estos pasos, se mostrará un mensaje de error como el de la figura 4-3, indicando qué pasos faltan por realizar.

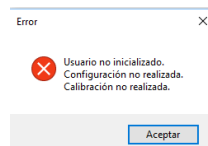


Figura 4-3: Error pasos previos al registro no realizados

El registro del usuario es simple, tal y como se puede apreciar en la figura 4-4. Se ha de introducir el nombre, los apellidos y la edad de la persona que vaya a realizar la prueba. El propio programa comprobará si el usuario ya había realizado alguna prueba anteriormente,

viendo los datos del fichero “*Doctor.txt*” situado en la carpeta “*Settings*” del directorio raíz de la aplicación y actualizará el valor al número de prueba correspondiente.

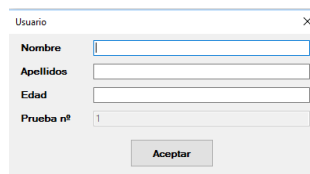
Un formulario de registro de usuario con el título "Usuario" y un botón de cerrar "X". Contiene cuatro campos de entrada: "Nombre", "Apellidos", "Edad" y "Prueba nº". El campo "Prueba nº" tiene el valor "1" predefinido. Debajo de los campos hay un botón "Aceptar".

Figura 4-4: Registro del usuario

Una vez introducidos los datos, se hará *click* en el botón “Aceptar”. Si falta algún dato se notificará con el mensaje de error mostrado en la figura 4-6. Y si se introduce una cadena no numérica y entera, se mostrará el mensaje de la figura 4-5, y se volverá al anterior número válido introducido, o tomará un valor por defecto, en caso de que no se hubiera introducido nada antes.

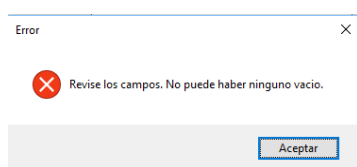
Un mensaje de error con el título "Error" y un botón de cerrar "X". Muestra un icono de error (una X roja) y el texto: "Revise los campos. No puede haber ninguno vacío." Debajo hay un botón "Aceptar".

Figura 4-6: Error faltan datos

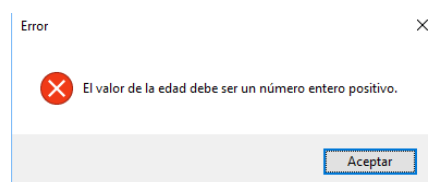
Un mensaje de error con el título "Error" y un botón de cerrar "X". Muestra un icono de error (una X roja) y el texto: "El valor de la edad debe ser un número entero positivo." Debajo hay un botón "Aceptar".

Figura 4-5: Error edad introducida no válida

La siguiente opción del menú es el botón de “Información”, el cual, al pulsarlo, abrirá una ventana en la que se resuelven las cuestiones: ¿Qué es un patrón de lectura?, ¿Para qué se utiliza?, ¿Cómo vamos a registrar los patrones de lectura?, ¿Qué incluye la etapa de configuración? y ¿Qué ocurre al pulsar en Registro de patrones?

Seguido de esta opción, tenemos el acceso al menú de configuración, mostrado en la figura 4-7. En esta podremos elegir una imagen en formato PNG, que será la imagen que se mostrará durante la prueba y también podremos elegir dónde se almacenarán los datos registrados. Una vez elegidos los directorios, se pulsará sobre el botón “Guardar y Continuar”.

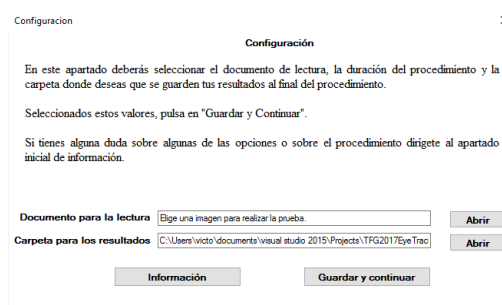
Una ventana de configuración con el título "Configuración" y un botón de cerrar "X". El contenido incluye: un texto explicativo sobre la selección de documento y carpeta; un texto indicando que se deben pulsar los botones "Guardar y Continuar" una vez seleccionados los valores; un texto advirtiendo de consultar la información inicial si hay dudas; dos campos de entrada con botones "Abrir" a la derecha: "Documento para la lectura" (con el texto "Elige una imagen para realizar la prueba") y "Carpeta para los resultados" (con la ruta "C:\Users\victo\documents\visual studio 2015\Projects\TFG2017\EyeTrac"); y dos botones "Información" y "Guardar y continuar" al final.

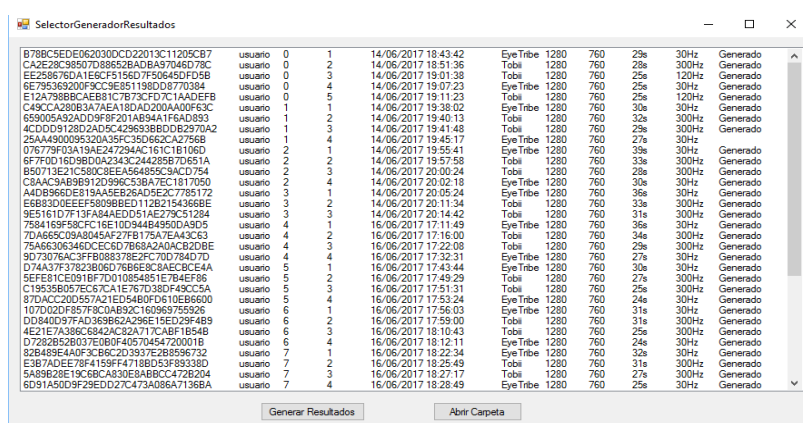
Figura 4-7: Configuración de la prueba

Una vez hecho esto, se tienen que elegir los dispositivos que se quieren utilizar para realizar la prueba, tanto el *eye-tracker* (pudiendo elegir entre usar el software de la marca Tobii o EyeTribe), como el de grabación de audio. Al pulsar en el botón de calibración, se abrirá el programa correspondiente al *eye-tracker* seleccionado y se procederá a la realización del calibrado (explicado en los Anexos C y D).

Al finalizar la calibración, ya estará todo listo para el registro de los patrones. Cuando el usuario esté listo, deberá pulsar en "Comenzar" (el cual ocupa toda la pantalla) y, entonces,

aparecerá la imagen elegida en la configuración a pantalla completa. Pasado el tiempo necesario para que los dispositivos se hayan inicializado, el usuario escuchará un pitido que indicará el comienzo del procedimiento de registro (grabando toda la actividad ocular y el audio). Para que este procedimiento sea correcto, es importante que el usuario comience a leer en voz alta el texto al escuchar dicha señal y, además, que intente mantener su concentración en la lectura del texto ya que cualquier distracción puede alterar los resultados.

Por último, está el menú de Generación de Resultados (mostrado en la figura 4-8). En este se muestra un listado de todas las pruebas con la información más relevante. Se indican el código SHA generado para la prueba (el cual es el nombre del directorio donde se almacenan los resultados), el nombre de los usuarios, el número de la prueba realizada, el momento en el que se realizó la prueba, el dispositivo que se utilizó, la resolución de la pantalla en la que se hizo la prueba, la frecuencia de muestreo que empleó el dispositivo y un indicador de si los resultados ya fueron generados o no.



SHA	usuario	ID	Fecha	Hora	Dispositivo	Resolución	Frecuencia	Estado
B78BC5E0E06203DCD22013C11205CB7	usuario	0	14/06/2017	18:43:42	EyeTrbe	1280	760	29s 30Hz Generado
CA2E28C98507D8852BADBA97046D78C	usuario	0	2	14/06/2017 18:51:36	Tobii	1280	760	28s 300Hz Generado
EE2586760A165C7F5156D7F56645D5F5B	usuario	0	3	14/06/2017 19:01:38	Tobii	1280	760	25s 120Hz Generado
6E795369200F3CC9E851198D0877D384	usuario	0	4	14/06/2017 19:07:23	EyeTrbe	1280	760	25s 30Hz Generado
E12A7988BCAEB81C7B73CFD7C1AADEFB	usuario	0	5	14/06/2017 19:11:23	Tobii	1280	760	25s 120Hz Generado
C49CC428063A7AE18DAD200A400F63C	usuario	1	1	14/06/2017 19:38:02	EyeTrbe	1280	760	30s 30Hz Generado
659005A92ADD9F8F201AB94A1F6AD893	usuario	1	2	14/06/2017 19:40:13	Tobii	1280	760	32s 300Hz Generado
4CDD09128D2AD5C4296938BD0B2970A2	usuario	1	3	14/06/2017 19:41:48	Tobii	1280	760	29s 300Hz Generado
25A449000952DA35FC350662CA2756B	usuario	1	4	14/06/2017 19:45:17	EyeTrbe	1280	760	27s 30Hz Generado
076779F03A19AE247294AC161C1B106D	usuario	2	1	14/06/2017 19:55:41	EyeTrbe	1280	760	33s 30Hz Generado
6F7FD16D98D0A2343C24428587D651A	usuario	2	2	14/06/2017 19:57:58	Tobii	1280	760	33s 300Hz Generado
B50713E2C580CEAE456485C3ACD754	usuario	2	3	14/06/2017 20:00:24	Tobii	1280	760	28s 300Hz Generado
C8AC3AB96912D996C53BA7EC1817050	usuario	2	4	14/06/2017 20:02:18	EyeTrbe	1280	760	30s 30Hz Generado
AAD896DE819AA5EB26AD5E2C7785172	usuario	3	1	14/06/2017 20:05:24	EyeTrbe	1280	760	36s 30Hz Generado
E5B3030EEF58096ED112821543668E	usuario	3	2	14/06/2017 20:11:34	Tobii	1280	760	33s 300Hz Generado
9E5161D7F13FA84AEDD51AE279C51284	usuario	3	3	14/06/2017 20:14:42	Tobii	1280	760	31s 300Hz Generado
7584169F59CFC15E10D944B4950DA8D5	usuario	4	1	16/06/2017 17:11:49	EyeTrbe	1280	760	36s 30Hz Generado
7DA65C209AB04FA27FB179A7E4A3C63	usuario	4	2	16/06/2017 17:16:00	Tobii	1280	760	34s 300Hz Generado
75A63306346DCEC6D7B68A2ADACB2D8E	usuario	4	3	16/06/2017 17:22:08	Tobii	1280	760	29s 300Hz Generado
9D73076AC3FFB088378E2FC70D784D7D	usuario	4	4	16/06/2017 17:32:31	EyeTrbe	1280	760	27s 30Hz Generado
D74A37F3782386D786693ACBCE4A	usuario	5	1	16/06/2017 17:43:44	EyeTrbe	1280	760	30s 30Hz Generado
5FEF81CE091BF7D010854851E7B4EF86	usuario	5	2	16/06/2017 17:49:29	Tobii	1280	760	27s 300Hz Generado
C19535B057EC57CA1E767D38DF9CC5A	usuario	5	3	16/06/2017 17:51:31	Tobii	1280	760	25s 300Hz Generado
97DACC200557A21ED5460F0610E86600	usuario	5	4	16/06/2017 17:53:24	EyeTrbe	1280	760	24s 30Hz Generado
107D02DF857F8C0AB92C160969755926	usuario	6	1	16/06/2017 17:56:03	EyeTrbe	1280	760	31s 30Hz Generado
DD840D97FAD359662A296E15ED29F4B9	usuario	6	2	16/06/2017 17:59:00	Tobii	1280	760	31s 300Hz Generado
4E21E7A386C0842AC2A717CA8F1854B	usuario	6	3	16/06/2017 18:10:43	Tobii	1280	760	25s 300Hz Generado
D7282628037E080F40570454720001B	usuario	6	4	16/06/2017 18:12:11	EyeTrbe	1280	760	24s 30Hz Generado
82B485E4A0F3CB6C2D3937E2B8596732	usuario	7	1	16/06/2017 18:22:34	EyeTrbe	1280	760	32s 30Hz Generado
E387ADCE78F4158FF47188D53838338D	usuario	7	2	16/06/2017 18:25:49	Tobii	1280	760	31s 300Hz Generado
5A89B28E19C6BCA830E8ABCC472B204	usuario	7	3	16/06/2017 18:27:17	Tobii	1280	760	27s 300Hz Generado
6D91A5D09F29EDD27C473A086A71368A	usuario	7	4	16/06/2017 18:28:49	EyeTrbe	1280	760	25s 30Hz Generado

Figura 4-8: Menú de generación de resultados

Para generar los resultados de una prueba, se deberá seleccionar la línea de la prueba en cuestión y se pulsará el botón “Generar Resultados”. Acto seguido, aparecerá una ventana como la mostrada en la figura 4-9 y, en el momento en que termine de crear todas las gráficas, vídeos y el fichero con los valores de la precisión de la prueba, se habilitará el botón “Abrir Carpeta Resultados” el cual, como indica su nombre, abrirá una ventana del explorador de archivos en la ubicación en la que se encuentran los resultados de la prueba.

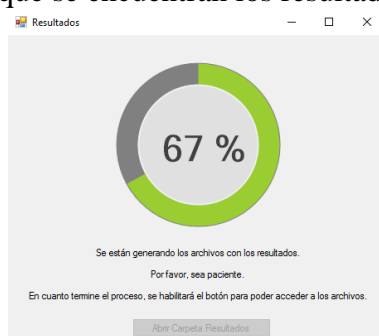


Figura 4-9: Generación de los resultados de una prueba

5 Integración, pruebas y resultados

5.1 Objetivos de la evaluación

El principal objetivo de la aplicación es evaluar la precisión de los dispositivos de rastreamiento de la actividad ocular durante una tarea de lectura orientada a la recomendación de diagnóstico de enfermedades como la dislexia y el déficit de atención, y poder comparar los resultados entre ellos. Para ello se ha diseñado una prueba que consiste en la lectura de dos textos que se leen dos veces, una vez con cada dispositivo.

5.2 La prueba

La prueba que se va a realizar para analizar la precisión de los dispositivos consiste en lo siguiente:

Los sujetos leerán dos textos dos veces, lo que hace un total de cuatro pruebas por usuario, dos por dispositivo. El motivo de realizar dos lecturas se debe a que, si el lector lee por segunda vez el mismo texto de manera consecutiva, puede afectar a su patrón de lectura. Por ello, la primera vez de cada texto se hace con un dispositivo distinto. Esto es:

- En la prueba 1 se lee el texto 1, grabando la actividad con el dispositivo 1.
- En la prueba 2 se lee el texto 1, grabando la actividad con el dispositivo 2.
- En la prueba 3 se lee el texto 2, grabando la actividad con el dispositivo 2.
- En la prueba 4 se lee el texto 2, grabando la actividad con el dispositivo 1.

Se ha decidido grabar el audio para poder tener una referencia acústica del proceso de lectura y tener otro método para evaluar si el punto en el que el usuario fijaba la mirada coincidía con lo que se estaba leyendo. La grabación también permite evaluar el movimiento de los ojos durante las pausas de la lectura.

5.2.1 Textos para la lectura

Ya que lo que importa en esta prueba es ver la precisión de los dispositivos, se han empleado textos simples y fáciles de leer. En concreto se han usado fragmentos de cuentos adaptados para niños de primaria (6 a 12 años de edad).

- Fichero 1: Caperucita roja:
<http://www.cuentosinfantilesadormir.com/cuento-caperucitaroja.htm>
- Fichero 2: Los tres cerditos:
<http://www.cuentosinfantilesadormir.com/cuento-lostrescerditos.htm>

Para ambos se han empleado las mismas características de diseño del texto:

- Tipo de letra: Arial.
- Colores: Texto negro y Fondo blanco
- Tamaño de fuente: 22.
- Interlineado 2.0

La uniformidad de los textos también facilita el análisis de precisión, puesto que las coordenadas de las líneas son las mismas.

5.3 Resultados

En total se han hecho observaciones sobre 13 sujetos, aunque tres de ellos no fueron capaces de completar la calibración del dispositivo de EyeTribe. Dos de ellos llevaban gafas y el tercero llevaba lentillas, y todos ellos tenían los ojos oscuros. Uno de ellos tenía déficit de atención, mientras que otro tenía estrabismo y el que usaba lentillas no tenía ningún tipo de patología que, a priori, pudiera afectar. No obstante, hubo otros sujetos con estrabismo que sí fueron capaces de realizar la calibración con los dos dispositivos. El sujeto con déficit de atención comentó que le costaba centrar la vista en la esfera que se movía en el proceso de calibración y que, cuando ya había realizado numerosos intentos, ya intentaba adelantarse al movimiento de la esfera y, aun así, el propio *software* de EyeTribe mostraba un mensaje indicando que no se había podido realizar la calibración como el que se ve en la figura 5-1.

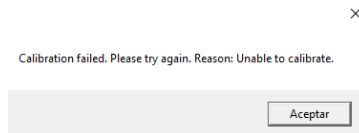


Figura 5-1: Error en el proceso de calibración de EyeTribe

Para el resto de participantes, se pudieron realizar todas las pruebas con los dos textos, aunque es cierto que en algunos casos las calibraciones no llegaron a niveles de calibración óptimos. Para estos usuarios que tuvieron malos resultados en la calibración no se pudo encontrar ningún patrón que explicara el motivo, unos llevaban gafas y tenían los ojos claros y otros no llevaban gafas y tenían los ojos oscuros. Lo que se pudo constatar fue el hecho de que un mismo sujeto con distintas gafas (unas con filtro de rayos azules y otras sin filtro) sí que había un cambio drástico a la hora de realizar las pruebas; con las gafas que contaban con el filtro salían unos resultados notablemente peores a los obtenidos con las que no lo tenían.

A continuación, se expondrán los resultados obtenidos con la aplicación, mostrándolos agrupados por usuario y texto para hacer la comparativa de los resultados de los dispositivos, haciendo una breve descripción inicial del usuario, mostrando las imágenes generadas y una tabla comparando las precisiones. Los resultados de las calibraciones de Tobii están guardados en forma de imágenes con los resultados de las pruebas, ya que no se pueden registrar como dato numérico. Debido al límite de páginas de esta memoria, se han seleccionado para su descripción los cinco ejemplos más significativos (se pueden encontrar todos los resultados en el enlace al código que figura en el apartado 4.2). No obstante, en el apartado 5.4, sí se han incluido en las tablas comparativas los resultados de todos los registros obtenidos. La elección de estos ejemplos se ha realizado teniendo en cuenta dos criterios: la calidad de las calibraciones y la diversidad de las cualidades de los usuarios (color de ojos, uso de gafas o lentillas, problemas de alguna índole). Por ello se han elegido a los usuarios 1, 3, 7, 10 y 11 para ser los mostrados en este apartado.

5.3.1 Usuario 1

El usuario contaba con 2 dioptrías de miopía y 1 dioptría de astigmatismo en ambos ojos. Usa gafas y tiene los ojos marrones. Se han incluido los resultados de este usuario puesto que era el mejor resultado de los usuarios que llevaban gafas.

5.3.1.1 Texto 1

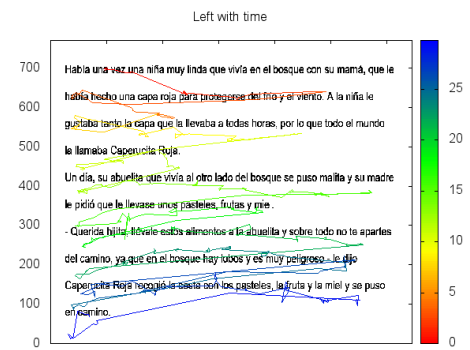


Figura 5-3: Registro usuario 1 texto 1 ojo izquierdo EyeTribe

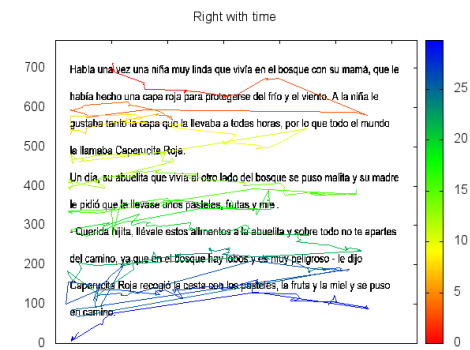


Figura 5-2: Registro usuario 1 texto 1 ojo derecho EyeTribe

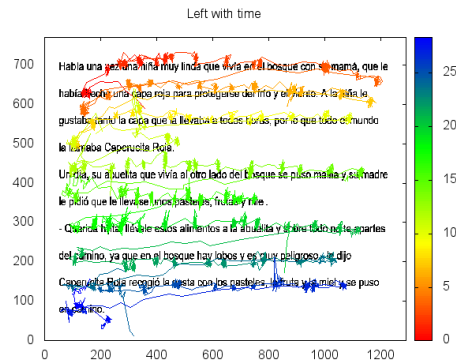


Figura 5-5: Registro usuario 1 texto 1 ojo izquierdo Tobii

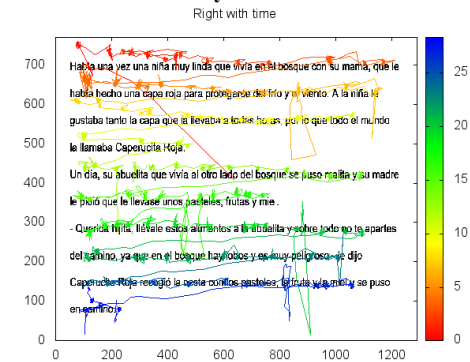


Figura 5-4: Registro usuario 1 texto 1 ojo derecho Tobii

5.3.1.2 Texto 2

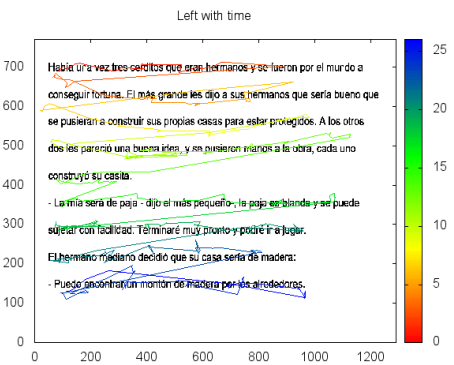


Figura 5-7: Registro usuario 1 texto 2 ojo izquierdo EyeTribe

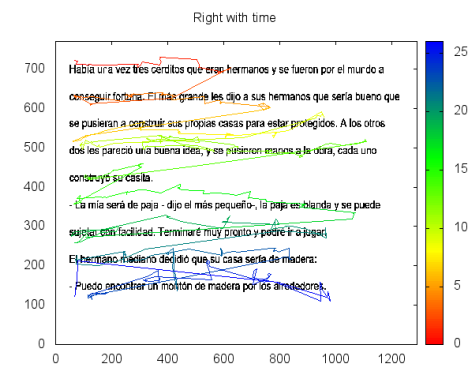


Figura 5-6: Registro usuario 1 texto 2 ojo derecho EyeTribe

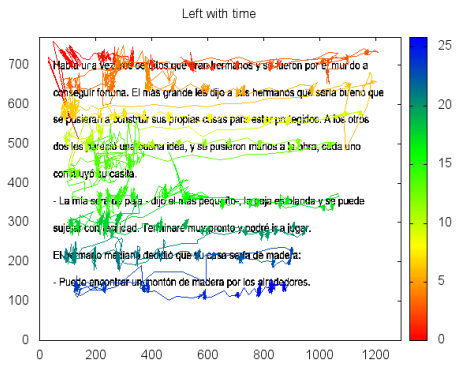


Figura 5-9: Registro usuario 1 texto 2 ojo izquierdo Tobii

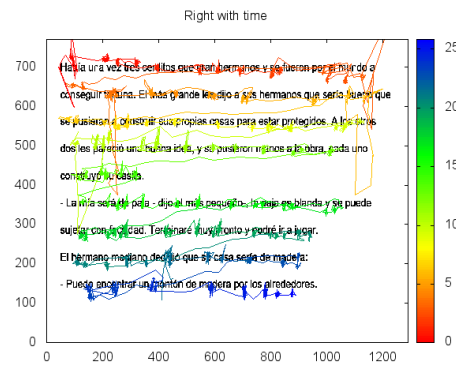


Figura 5-8: Registro usuario 1 texto 2 ojo izquierdo Tobii

5.3.2 Usuario 3

Este usuario no usa gafas y obtuvo unos resultados en la calibración muy buenos en ambos dispositivos.

5.3.2.1 Texto 1

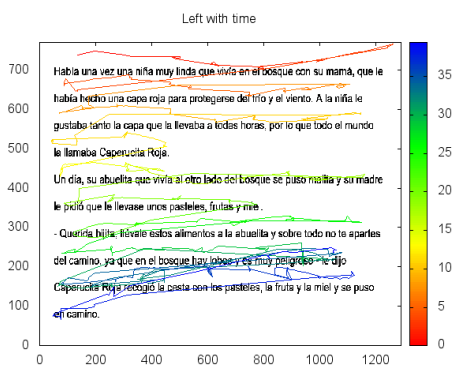


Figura 5-11: Resultados usuario 3 texto 1 ojo izquierdo EveTribe

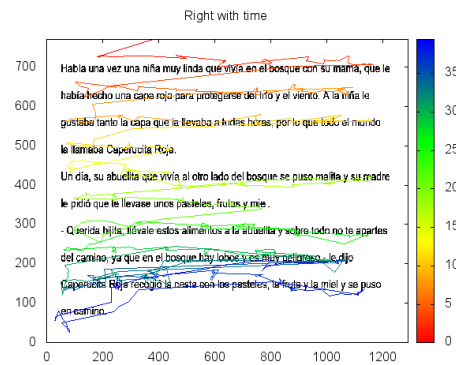


Figura 5-10: Resultados usuario 3 texto 1 ojo derecho EyeTribe

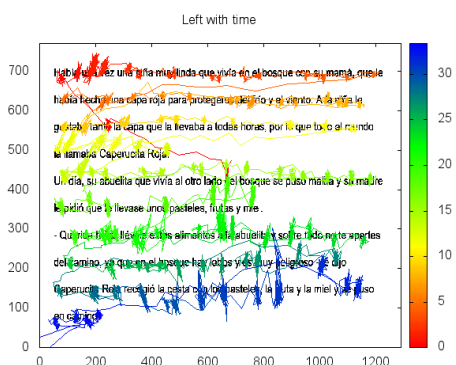


Figura 5-13: Resultados usuario 3 texto 1 ojo izquierdo Tobii

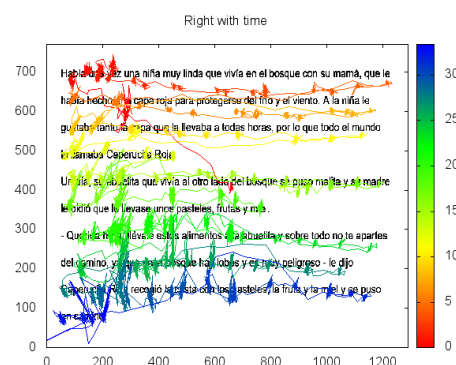


Figura 5-12: Resultados usuario 3 texto 1 ojo derecho Tobii

5.3.2.2 Texto 2

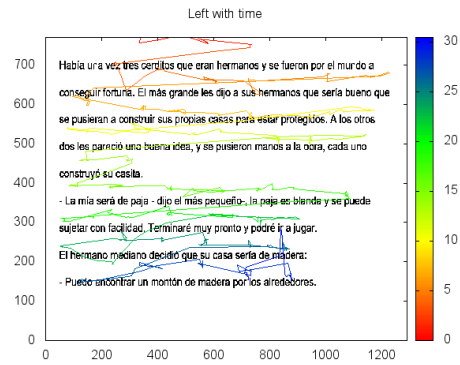


Figura 5-14: Resultados usuario 3 texto 2 ojo izquierdo
EyeTribe

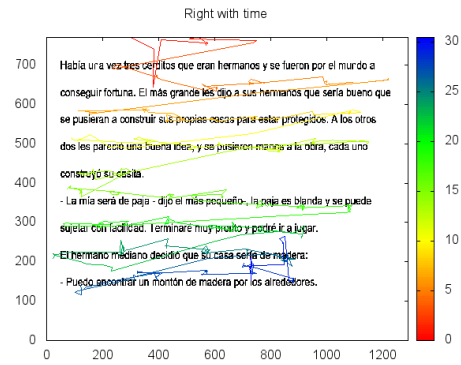


Figura 5-15: Resultados usuario 3 texto 2 ojo derecho
EyeTribe

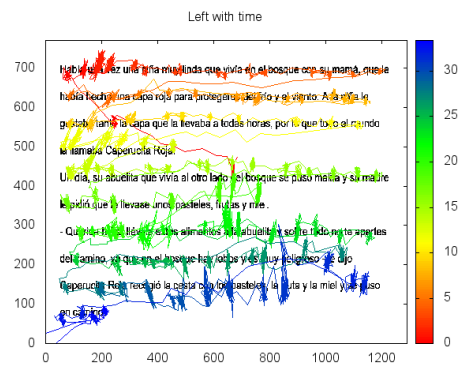


Figura 5-16: Resultados usuario 3 texto 2 ojo izquierdo
Tobii

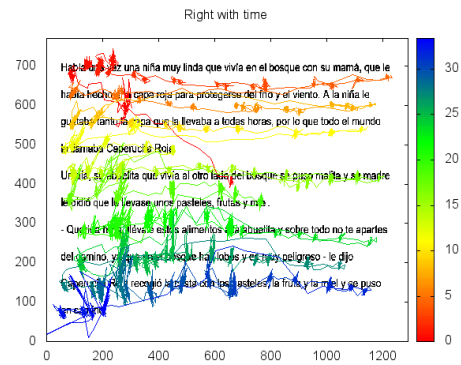


Figura 5-17: Resultados usuario 3 texto 2 ojo derecho
Tobii

5.3.3 Usuario 7

Este usuario no usa gafas y obtuvo unos resultados en la calibración muy buenos en ambos dispositivos. A pesar de haber tenido tan buenos resultados en la calibración, este es un ejemplo de que los dispositivos pueden hacer un mal registro en algunas pruebas (especialmente las del dispositivo de EyeTribe).

5.3.3.1 Texto 1

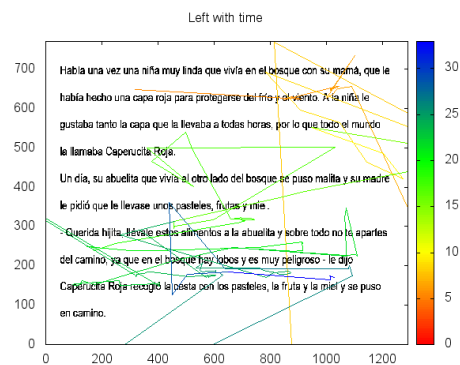


Figura 5-18: Resultados usuario 7 texto 1 ojo izquierdo
EyeTribe

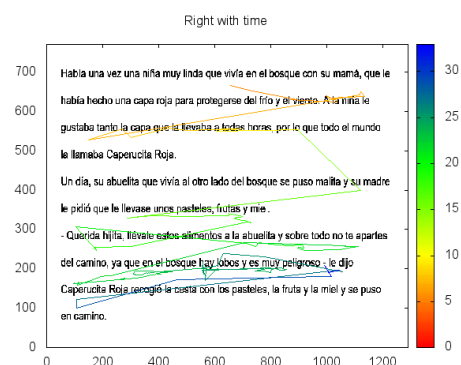


Figura 5-19: Resultados usuario 7 texto 1 ojo derecho
EyeTribe

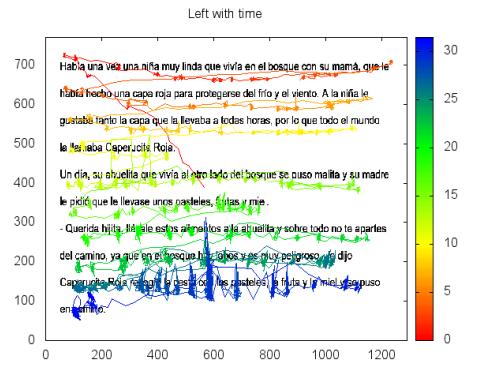


Figura 5-21: Resultados usuario 7 texto 1 ojo derecho Tobii

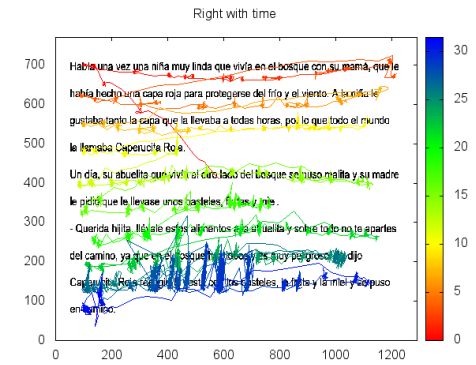


Figura 5-20: Resultados usuario 7 texto 1 ojo derecho Tobii

5.3.3.1 Texto 2

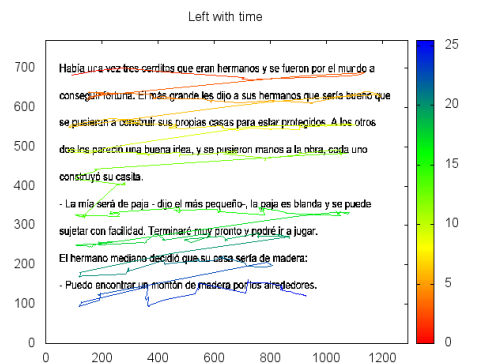


Figura 5-23: Resultados usuario 7 texto 2 ojo izquierdo EyeTribe

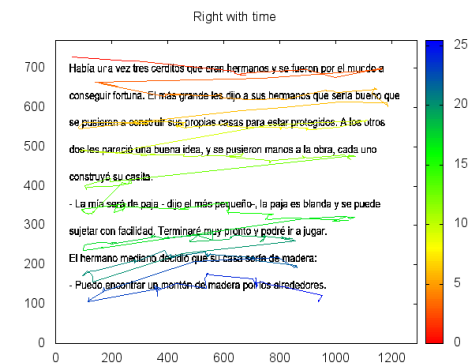


Figura 5-22: Resultados usuario 7 texto 2 ojo derecho EyeTribe

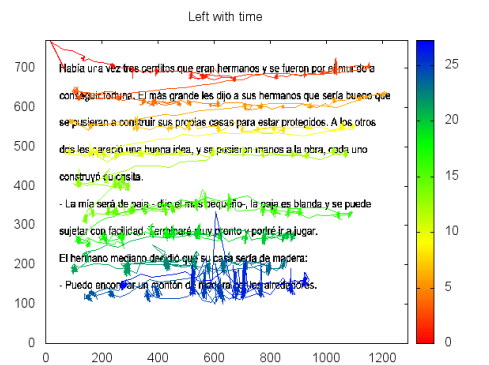


Figura 5-24: Resultados usuario 7 texto 2 ojo izquierdo Tobii

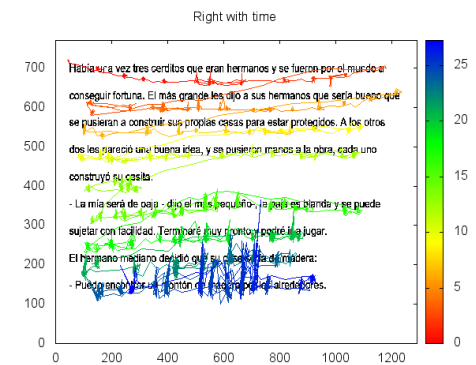


Figura 5-25: Resultados usuario 7 texto 2 ojo derecho Tobii

5.3.4 Usuario 10

Este usuario no usa gafas y obtuvo unos resultados en la calibración muy buenos en ambos dispositivos.

5.3.4.1 Texto 1

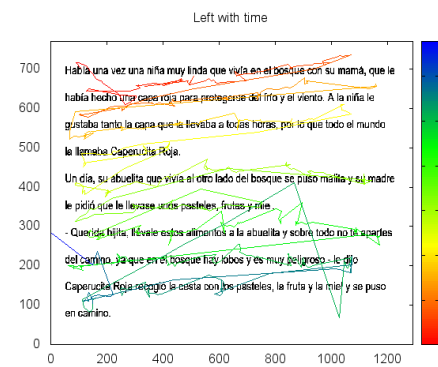


Figura 5-27: Resultados usuario 10 texto 1 ojo izquierdo EyeTribe

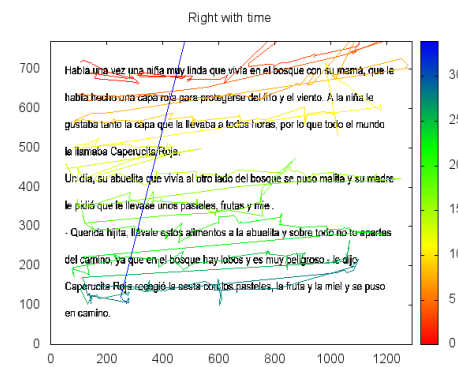


Figura 5-26: Resultados usuario 10 texto 1 ojo derecho EyeTribe

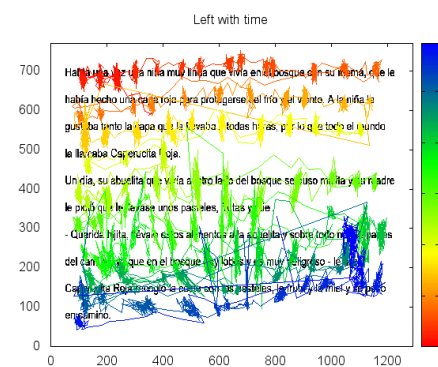


Figura 5-29: Resultados usuario 10 texto 1 ojo izquierdo Tobii

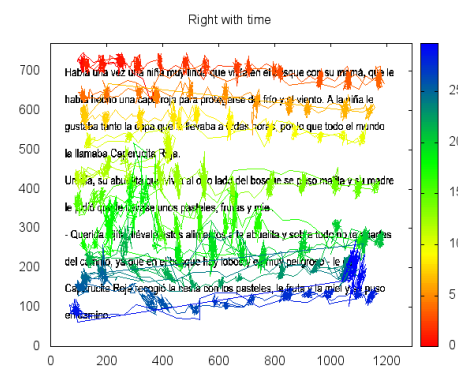


Figura 5-28: Resultados usuario 10 texto 1 ojo derecho Tobii

5.3.4.2 Texto 2

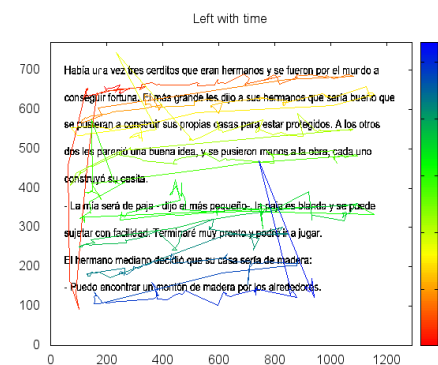


Figura 5-31: Resultados usuario 10 texto 2 ojo izquierdo EyeTribe

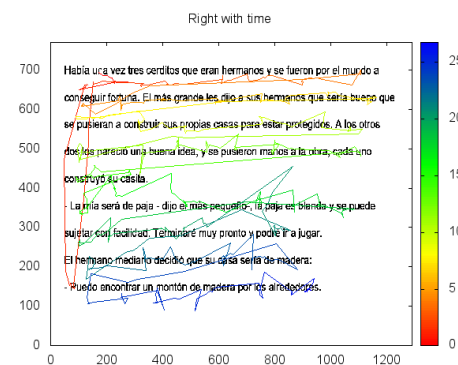


Figura 5-30: Resultados usuario 10 texto 2 ojo derecho EyeTribe

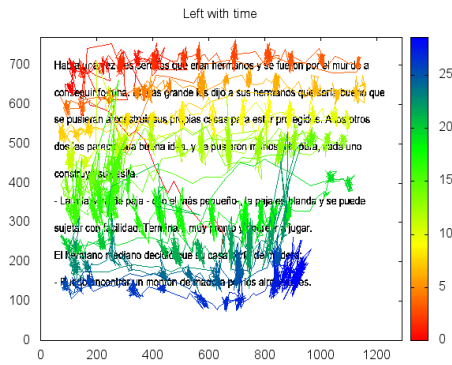


Figura 5-32: Resultados usuario 10 texto 2 ojo izquierdo Tobii

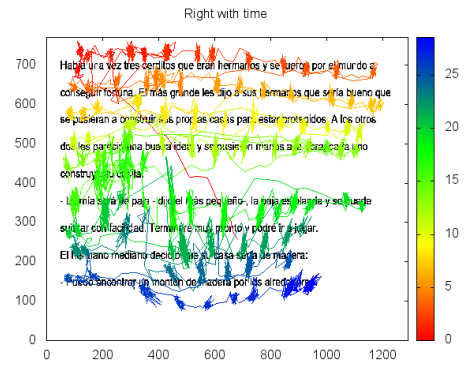


Figura 5-33: Resultados usuario 10 texto 2 ojo derecho Tobii

5.3.5 Usuario 11

Este usuario no usa gafas y obtuvo unos resultados en la calibración muy buenos en ambos dispositivos.

5.3.5.1 Texto 1

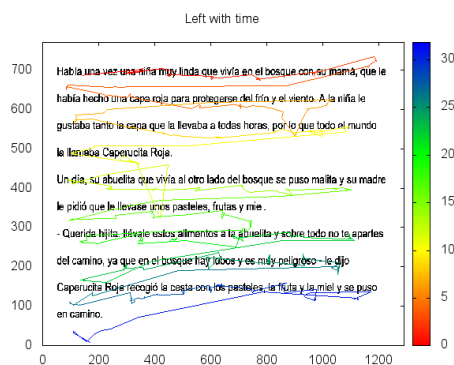


Figura 5-35: Resultados usuario 11 texto 1 ojo izquierdo EyeTribe

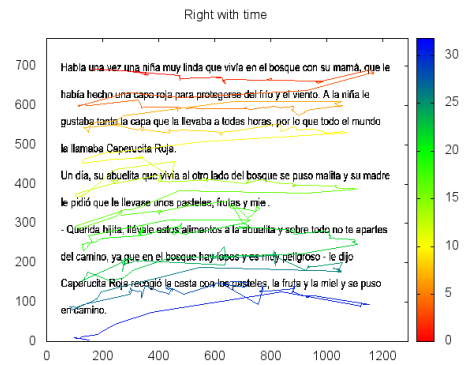


Figura 5-34: Resultados usuario 11 texto 1 ojo derecho EyeTribe

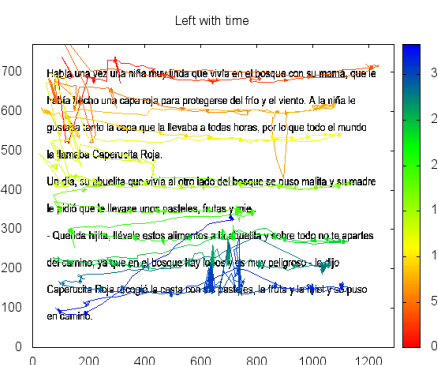


Figura 5-37: Resultados usuario 11 texto 1 ojo izquierdo Tobii

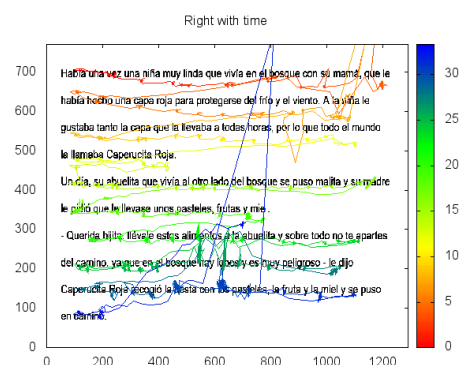


Figura 5-36: Resultados usuario 11 texto 1 ojo derecho Tobii

5.3.5.2 Texto 2

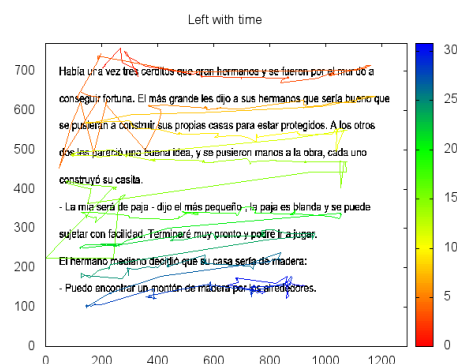


Figura 5-39: Resultados usuario 11 texto 2 ojo izquierdo EyeTribe

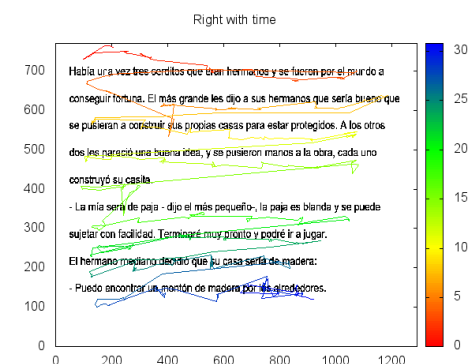


Figura 5-38: Resultados usuario 11 texto 2 ojo derecho EyeTribe

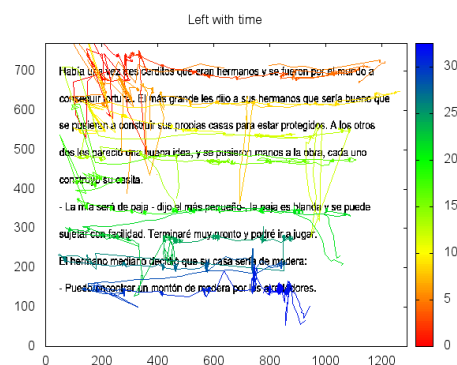


Figura 5-40: Resultados usuario 11 texto 2 ojo izquierdo Tobii

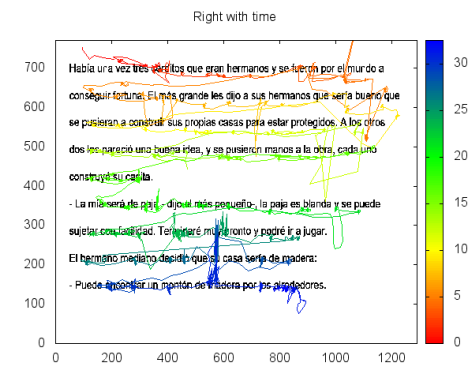


Figura 5-41: Resultados usuario 11 texto 2 ojo derecho Tobii

5.4 Análisis de la precisión

Para ver cómo se han realizado los cálculos de precisión y desviación media de los fallos, se puede ir al Anexo F, en el que se describe detalladamente el proceso. En las siguientes tablas se detallan los resultados de la precisión de las pruebas, junto con la desviación media registrada. Para no ensuciar los resultados, se han usado sólo aquellos que hubieran hecho una calibración que fuera al menos de 4 estrellas para el dispositivo de EyeTribe y no hubiera mucha desviación en la calibración del dispositivo de Tobii (Para poder compararlo con el de EyeTribe, se establecieron unas referencias para definir lo que era una puntuación de 5 estrellas, que viene siendo que todas las fijaciones estaban en los puntos que debían o se salían mínimamente, como se puede ver en la figura 0-9 del Anexo D y, a partir de ahí, según iba empeorando el resultado se clasificaron con más o menos puntos).

	Puntuación calibración		Precisión (%)		Desviación Media en los fallos (Píxeles)	
	Tobii	EyeTribe	Tobii	EyeTribe	Tobii	EyeTribe
Usuario 1	4	4	89,22	86,17	25,74	41,86
Usuario 3	5	5	89,78	84,07	28,40	35,28
Usuario 4	5	5	97,35	2,28	21,87	508,15
Usuario 6	4	4	90,29	99,33	49,89	117,16
Usuario 7	5	5	95,18	6,28	34,74	276,53
Usuario 9	4	4	91,50	81,25	55,12	75,78

Usuario 10	5	5	63,44	84,04	57,65	30,06
Usuario 11	5	5	94,60	89,90	44,01	24,54

Tabla 5-1: Resultados texto 1 ojo izquierdo

	Puntuación calibración		Precisión (%)		Desviación Media en los fallos (Píxeles)	
	Tobii	EyeTribe	Tobii	EyeTribe	Tobii	EyeTribe
Usuario 1	4	4	95,90	85,48	25,45	68,09
Usuario 3	5	5	86,67	97,59	28,45	27,22
Usuario 4	5	5	87,23	84	42,51	124,63
Usuario 6	4	4	73,88	94,91	38,68	112,27
Usuario 7	5	5	93,46	0,44	26,06	330,24
Usuario 9	4	4	88,33	84,65	45,70	45,66
Usuario 10	5	5	53,81	35,50	112,93	221,02
Usuario 11	5	5	96,59	97,33	30,73	30,69

Tabla 5-2: Resultados texto 1 ojo derecho

	Puntuación calibración		Precisión (%)		Desviación Media en los fallos (Píxeles)	
	Tobii	EyeTribe	Tobii	EyeTribe	Tobii	EyeTribe
Usuario 1	4	4	95,90	85,48	23,75	175,72
Usuario 3	5	5	96,52	82,54	24,21	54,84
Usuario 4	5	5	94,47	99,75	28,01	232,13
Usuario 6	4	4	94,24	93,17	50,67	31,06
Usuario 7	5	5	98,52	99,12	23,54	23,23
Usuario 9	4	4	88,63	3,71	31,00	221,23
Usuario 10	5	5	60,77	88,43	66,85	33,24
Usuario 11	5	5	76,94	98,04	31,16	26,66

Tabla 5-3: Resultados texto 2 ojo izquierdo

	Puntuación calibración		Precisión (%)		Desviación Media en los fallos (Píxeles)	
	Tobii	EyeTribe	Tobii	EyeTribe	Tobii	EyeTribe
Usuario 1	4	4	99,12	89,59	19,87	31,62
Usuario 3	5	5	81,73	78,84	34,53	27,41
Usuario 4	5	5	86,95	98,76	41,35	188,40
Usuario 6	4	4	93,46	97,21	25,41	37,50
Usuario 7	5	5	95,70	3,52	28,59	252,70
Usuario 9	4	4	67,46	90,03	64,12	50,72
Usuario 10	5	5	57,87	78,10	53,20	32,77
Usuario 11	5	5	80,98	12,65	32,62	239,18

Tabla 5-4: Resultados texto 2 ojo derecho

En vista de las anteriores tablas, se obtiene que el promedio de precisión de Tobii es de un 86,14 % y el de EyeTribe es de un 72,26 %. Aunque con el dato previo pueda parecer que los resultados no han sido muy dispares, la desviación media de Tobii ha sido de 38,96 píxeles, mientras que la de EyeTribe ha sido de 116,48 píxeles.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

El objetivo de este trabajo es analizar la diferencia de precisión de distintos dispositivos de seguimiento ocular. Para ello se ha realizado una aplicación que, a partir de una prueba de lectura, genera una serie de gráficos que muestran los patrones de lectura junto con información adicional que permite conocer la calidad de los resultados. A la hora de implementar la aplicación, se ha pensado en su posible ampliación y uso en distintos ordenadores y dispositivos de seguimiento ocular (siempre y cuando utilicen el sistema operativo de Windows), por lo que se ha realizado todo de manera genérica.

En vista a los resultados expuestos en el apartado 5.3, se observa que hay una diferencia no muy grande en el cómputo global en cuanto a lo que se refiere a la precisión de posiciones coherentes. Aunque se ve que el dispositivo de Tobii es mucho más constante en la calidad de sus resultados, a excepción de algunos casos concretos. Asimismo, hay una diferencia muy grande entre las desviaciones medias de los fallos, lo cual es un dato muy importante, ya que el dispositivo de EyeTribe no solo falla más, sino que también lo hace por mucho más.

También se ha podido observar que, en ocasiones, independientemente de haber realizado una buena calibración, los resultados del seguimiento del patrón de lectura no son óptimos, esto puede haberse debido a que los sujetos se hayan movido entre el momento de la calibración y la realización de la prueba (aunque en principio ambos dispositivos funcionan con tolerancia a ligeros movimientos).

Asimismo, se ha observado que el dispositivo de EyeTribe era más complejo de colocar en una posición que detectara bien los ojos de los usuarios. También, en numerosas ocasiones daba problemas ya que pedía la actualización de software a mitad de la calibración, incluso habiendo desactivado la opción de buscar nuevo software y teniendo la red apagada. Para solventar el problema había que reiniciar el ordenador. El dispositivo Tobii Tx-300, en ocasiones tampoco era capaz de realizar la calibración correctamente, e incluso hubo momentos en los que sólo era capaz de detectar un único ojo. Eso sí, no era preciso reiniciar la computadora para que solucionara el problema, se corregía el problema al cabo de unos segundos.

Otro problema que tienen ambos dispositivos es el sobrecalentamiento, sobre todo en los días de calor, cuando se realizaron las pruebas con los voluntarios. Al llevar un tiempo haciendo pruebas, los dispositivos se calentaban mucho, también el dispositivo de Tobii x2-30 que se utilizó inicialmente en este proyecto. El Tx-300 tiene un sistema de ventilación integrado y no desprendía tanto calor como los otros. Además, el problema del calor hacía que al cabo de un rato de uso (algo más de una hora) fuera conveniente desconectarlo y dejar que se enfriara (incluso teniendo en cuenta que el dispositivo permanecía apagado cuando no estaba siendo usado).

Con todo ello se puede concluir que, para la detección de patrones de lectura, puede llegar a ser suficiente un dispositivo de bajo coste como el EyeTribe, teniendo en cuenta que puede dar lugar a bastantes problemas, haciendo imposible su uso para algunos usuarios. Es necesario mejorar la fiabilidad en los dispositivos de bajo coste, que por otro lado son necesarios para la diseminación de las aplicaciones biomédicas que utilizan seguimiento

ocular. No obstante, sería recomendable hacer uso de algún dispositivo de una gama superior para validar los resultados de los dispositivos de bajo coste, que grabe a una mayor frecuencia y con una menor tasa de fallos, pero sin ser necesario dar el salto a un dispositivo de gama alta como es el Tobii Tx-300 ya que la diferencia de precio es muy significativa.

Adicionalmente, se ha podido observar, gracias a la inclusión del audio en los videos resultantes, que la mirada va siempre por delante que la voz.

6.2 Trabajo futuro

Este proyecto ha sido diseñado de manera genérica para que pueda funcionar con cualquier dispositivo eye-tracker que tenga un SDK compatible con .Net, siendo necesario añadir sólo unas pocas líneas de código. Además, al estar preparado para cualquier dimensión de pantalla, no será necesario hacer adaptaciones de las resoluciones. Esto hace que las posibilidades de ampliar la aplicación sean muy grandes.

Con una ampliación se podría hacer una comparativa entre un mayor número de dispositivos, así como diseñar distintos tipos de pruebas para evaluar el comportamiento de estos en distintos escenarios. Al tener ya integrada la lógica de la grabación del seguimiento ocular y almacenando los puntos en un fichero, realmente se puede adaptar el código para poner a prueba este tipo de aparatos. Para generalizar su uso en cualquier aplicación de seguimiento ocular, sólo habría que, en lugar de mostrarse de fondo de pantalla una imagen para leer, se pusiera cualquier otra prueba.

En un futuro se podría integrar en esta aplicación un conjunto de algoritmos de aprendizaje automático, en particular aquellos que pueden tratar las series multivariadas que produce el seguimiento ocular (Rozado et al., 2010, 2011, 2012a, 2012b, 2012c, 2015), con el objetivo de clasificar automáticamente los registros de lectura en términos de personas sanas o personas a las que se recomienda un diagnóstico médico.

También podría ser interesante hacer otros tipos de mediciones complementarias (como, por ejemplo, la de la actividad cerebral mediante el uso de EEG) durante las pruebas para realizar estudios cognitivos más completos. Al tener información de la actividad del cerebro, la dilatación de las pupilas y la actividad ocular, junto con la información de la actividad realizada, la aplicación podría usarse en muchos tipos de estudios biomédicos.

Referencias

- almajjer, E. S., Mathôt, S., and der Stigchel, S. (2014). PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. *Behav. Res. Methods* 46, 913–921. doi:10.3758/s13428-013-0422-2.
- Ferhat, O., and Vilarino, F. (2016). Low Cost Eye Tracking: The Current Panorama. *Comput. Intell. Neurosci.* 2016, 8680541.
- Funke, G., Greenlee, E., Carter, M., Dukes, A., Brown, R., and Menke, L. (2016). Which Eye Tracker Is Right for Your Research? Performance Evaluation of Several Cost Variant Eye Trackers. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 60, 1240–1244. doi:10.1177/1541931213601289.
- Holzman, P. S., Proctor, L. R., and Hughes, D. W. (1973). Eye-Tracking Patterns in Schizophrenia. *Science* (80-.). 181, 179–181. doi:10.1126/science.181.4095.179.
- Jacobsen, L. K., Hong, W. L., Hommer, D. W., Hamburger, S. D., Castellanos, F. X., Frazier, J. A., et al. (2017). Smooth pursuit eye movements in childhood-onset schizophrenia: Comparison with attention-deficit hyperactivity disorder and normal controls. *Biol. Psychiatry* 40, 1144–1154. doi:10.1016/S0006-3223(95)00630-3.
- Kretschmar, F., Pleimling, D., Hosemann, J., Füssel, S., Bornkessel-Schlesewsky, I., and Schlewsky, M. (2013). Subjective Impressions Do Not Mirror Online Reading Effort: Concurrent EEG-Eyetracking Evidence from the Reading of Books and Digital Media. *PLoS One* 8, e56178.
- McClatchy, S. (1998). In tracking Net, the eyes have it. *Regist.*, 12/24. Available at: <https://news.google.com/newspapers?id=yngVAAAAIABAJ&sjid=k-sDAAAIAAJ&pg=6872,4783461&dq=eye-tracking&hl=en> [Accessed March 20, 2017].
- Pereira, M. L. G. F., Camargo, M. von Z. A., Aprahamian, I., and Forlenza, O. V (2014). Eye movement analysis and cognitive processing: detecting indicators of conversion to Alzheimer's disease. *Neuropsychiatr. Dis. Treat.* 10, 1273–1285. doi:10.2147/NDT.S55371.
- Rayner, K. (1978). Eye movements in reading and information processing. *Psychol. Bull.* 85, 618.
- Rayner, K. (1979). Eye guidance in reading: Fixation locations within words. *Perception* 8, 21–30.
- Reyes Gentil, M., and Varona Martínez, P. (2017). REGISTRO DE PATRONES DE LECTURA CON DISPOSITIVOS DE EYE TRACKER DE BAJO COSTE Y ESTUDIO DE SU APLICACIÓN PARA LA RECOMENDACIÓN DE DIAGNÓSTICO DE PATOLOGÍAS. 116.
- Rozado, D., Agustin, J. S., Rodriguez, F. B., and Varona, P. (2012a). Gliding and saccadic gaze gesture recognition in real time. *ACM Trans. Interact. Intell. Syst.* 1, 10:1--10:27. doi:10.1145/2070719.2070723.
- Rozado, D., Moreno, T., San Agustin, J., Rodriguez, F. B. B., and Varona, P. (2015). Controlling a Smartphone Using Gaze Gestures as the Input Mechanism. *Human-Computer Interact.* 30, 34–63. doi:10.1080/07370024.2013.870385.
- Rozado, D., Rodriguez, F. B., and Varona, P. (2010). Optimizing Hierarchical Temporal Memory for Multivariable Time Series. *Lect. Notes Comput. Sci.* 6353, 506–518. doi:10.1007/978-3-642-15822-3_62.
- Rozado, D., Rodriguez, F. B., and Varona, P. (2011). Gaze Gesture Recognition with Hierarchical Temporal Memory Networks. *Lect. Notes Comput. Sci.* 6691, 1–8. doi:10.1007/978-3-642-21501-8_1.

- Rozado, D., Rodriguez, F. B., and Varona, P. (2012b). Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition. *Neurocomputing* 79, 75–86. doi:10.1016/j.neucom.2011.10.005.
- Rozado, D., Rodriguez, F. B., and Varona, P. (2012c). Low cost remote gaze gesture recognition in real time. *Appl. Soft Comput.* 12, 2072–2084. doi:10.1016/j.asoc.2012.02.023.
- Slykhuis, D. A., Wiebe, E. N., and Annetta, L. A. (2005). Eye-Tracking Students' Attention to PowerPoint Photographs in a Science Education Setting. *J. Sci. Educ. Technol.* 14, 509–520. Available at: <http://www.jstor.org/stable/40186731>.
- Tobii (2012). Tobii ® Technology Products & Services – Price list.
- Tobii ® Technology Accuracy and precision test method for remote eye trackers (2011). Available at: <http://www.tobii.com/siteassets/tobii-pro/learn-and-support/use/what-affects-the-performance-of-an-eye-tracker/tobii-test-specifications-accuracy-and-precision-test-method.pdf?v=2.1.1> [Accessed March 7, 2017].
- Tobii TX300 Eye Tracker Available at: <http://www.tobii.com/siteassets/tobii-pro/product-descriptions/tobii-pro-tx300-product-description.pdf?v=1.0> [Accessed March 9, 2017].
- Zawoyski, A. M., Ardoin, S. P., and Binder, K. S. (2015). Using Eye Tracking to Observe Differential Effects of Repeated Readings for Second-Grade Students as a Function of Achievement Level. *Read. Res. Q.* 50, 171–184. doi:10.1002/rrq.91.

Glosario

SDK	Software Development Kit
SHA	Secure Hash Algorithm
PNG	Portable Network Graphics
ECG	Electrocardiogramas
EEG	Electroencefalogramas

Anexos

A Manual de instalación

Para la instalación del dispositivo de EyeTribe, lo único necesario es descargar e instalar el software de la página oficial, el cual redirige al repositorio de Github <https://github.com/EyeTribe/sdk-installers>.

En el caso de Tobii, el x2-30 no necesita ningún tipo de instalación. Pero, con el Tx-300, es necesario hacer una configuración de las opciones de internet para que el ordenador lo pueda reconocer, puesto que el dispositivo se conecta vía ethernet. Siguiendo los siguientes puntos, extraídos del manual de usuario del dispositivo, debería ser suficiente para que funcione:

1. Conectar el cable ethernet tanto al dispositivo como al ordenador.
2. Entrar al “Centro de redes y recursos compartidos”.
3. Pinchar sobre el icono de conexión de ethernet.
4. Clicar el botón de propiedades.
5. Pinchar la opción “Protocolo de Internet versión 4 (TCP/IPv4)” y clicar el botón de Propiedades.
6. Asegurarse de que la opción “Obtener una dirección IP automáticamente” está marcada y pulsar OK.
7. Aplicar cambios y reiniciar el ordenador con el *eye-tracker* conectado y encendido.

Al terminar el reinicio, el ordenador ya debería ser capaz a reconocer el dispositivo. Un indicador de que lo está haciendo es que, en el icono de la conexión de ethernet, debe aparecer el icono de conectividad limitada.

B Diagrama de Clases

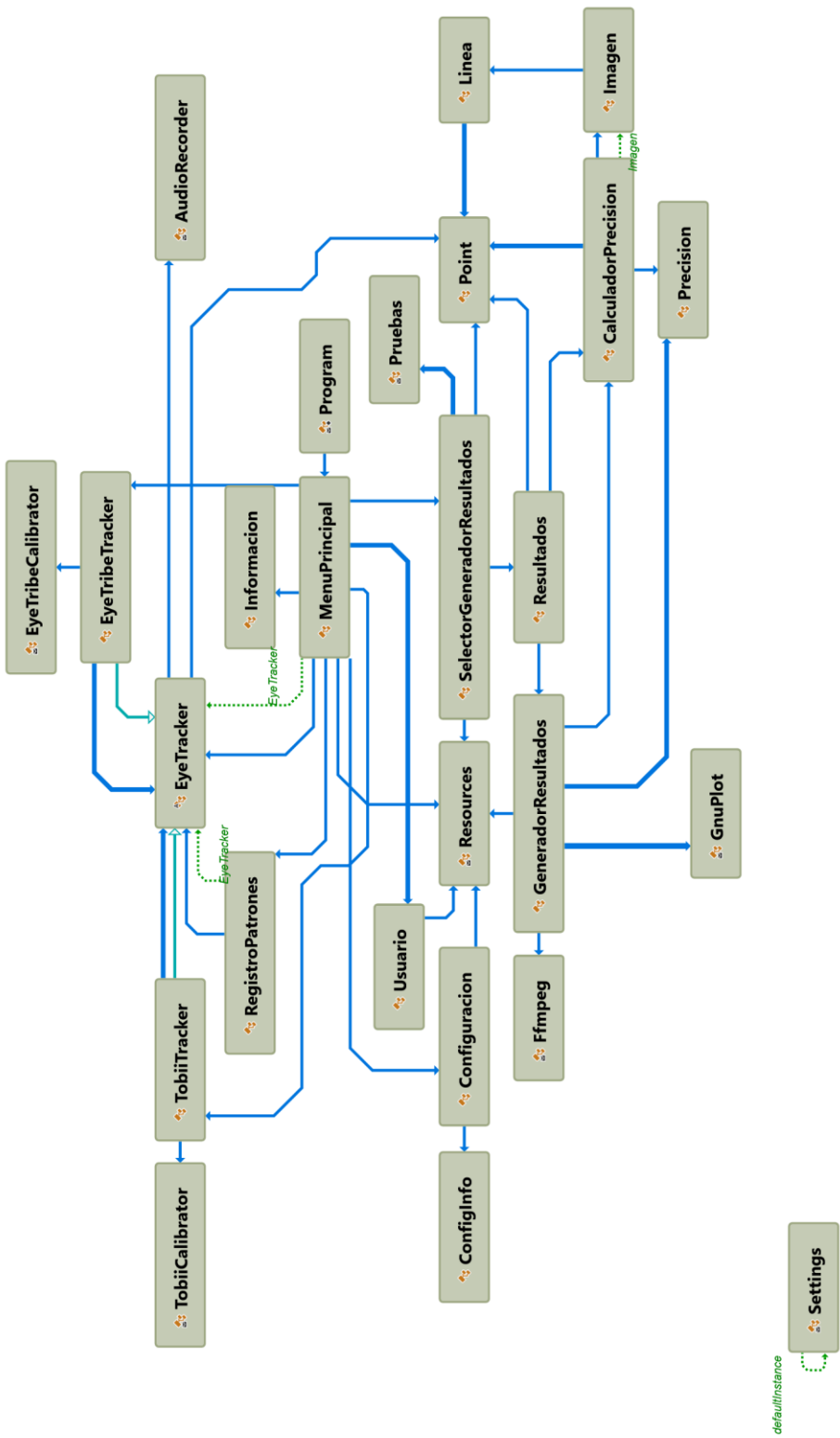


Figura 0-1: Diagrama de clases de la aplicación

C Calibración del dispositivo de TheEyeTribe

La calibración del dispositivo de TheEyeTribe, se realiza desde la aplicación oficial de la marca. Al comenzar lo más probable es que aparezca el icono rojo con los ojos tachados, indicándonos que el dispositivo no está encontrando nuestros ojos, tal y como se ve en la figura 0-2. Si nos encontramos en esta situación, lo que se deberá hacer es colocar el dispositivo en una posición en la que sea capaz de vernos.

Cuando el dispositivo reconozca al usuario, aparecerán unos ojos que se mueven según nuestros movimientos, sobre un fondo verde (como en la figura 0-3), el cual puede variar la tonalidad acercándose al color rojo cuanto peor sea la detección.



Figura 0-2: Ojos no detectados EyeTribe



Figura 0-3: Ojos detectados EyeTribe

En este momento, podemos iniciar la calibración pulsando el botón “*Calibrate*”. A continuación, se mostrará sobre un fondo gris un círculo, el cual hay que seguir con la mirada. Al cabo de unos segundos, se finalizará el proceso y mostrará una pantalla como la de la figura 0-4, en la que se muestra, con un sistema de medición de un máximo de cinco estrellas, la calidad de la calibración, siendo 0 estrellas que hay que repetir la calibración y cinco estrellas una calibración excelente. En la imagen de ejemplo se ve una calibración “moderada”, la cual sería conveniente repetir, para tener unos resultados mejores. Lo óptimo sería trabajar con calibraciones de al menos cuatro estrellas.

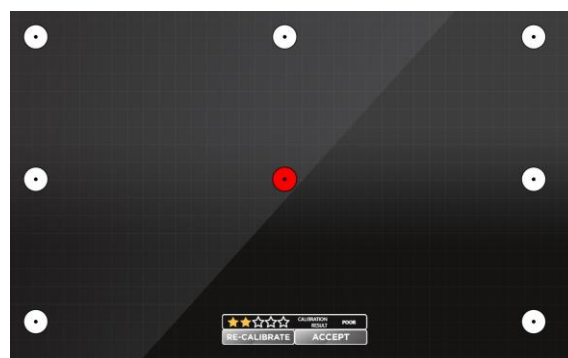


Figura 0-4: Resultado calibración

Además, la aplicación nos permite elegir el número de puntos que se quieren usar en la calibración. Teóricamente, a mayor número de puntos mejor han de ser los resultados. También se pueden cambiar otras configuraciones como los colores de la pantalla de

calibración, sobre qué pantalla se quiere hacer o, en la pestaña de opciones permite realizar un muestreo, y generar un fichero con los datos que devuelve el dispositivo.

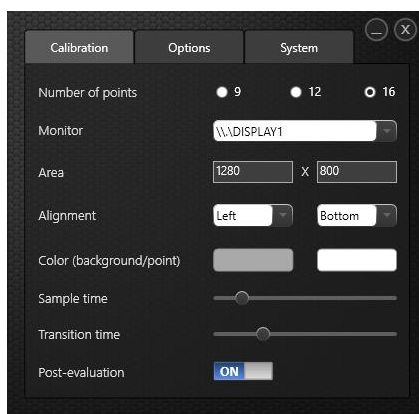


Figura 0-5: Panel de configuración de la aplicación de EyeTribe

D Calibración del dispositivo de Tobii

Al abrirse la aplicación, se muestra un menú en el que se permite seleccionar los dispositivos de Tobii, tal y como se ve en la figura 0-6.

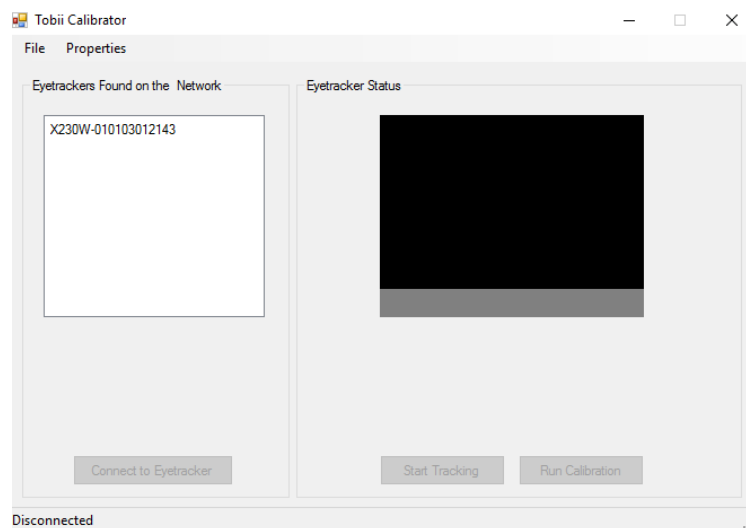


Figura 0-6: Aplicación calibración Tobii

Seleccionando el dispositivo que queremos conectar y pulsando el botón “*Connect to Eyetracker*”, se habilitarán los botones “*Start Recording*” y “*Run Calibration*”. Es conveniente pulsar primero el botón que comienza la grabación, pues así se puede saber si el dispositivo está bien colocado y reconoce al individuo que esté situado frente a él. En la figura 0-7 se puede observar lo que ocurre cuando el dispositivo no detecta ningún ojo, mientras que en la figura 0-8 si los detecta.

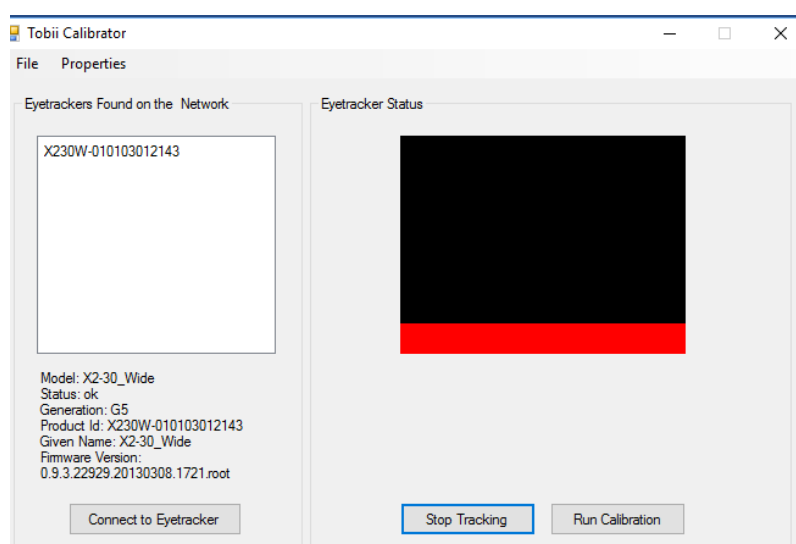


Figura 0-7: Tobii no reconoce ojos.

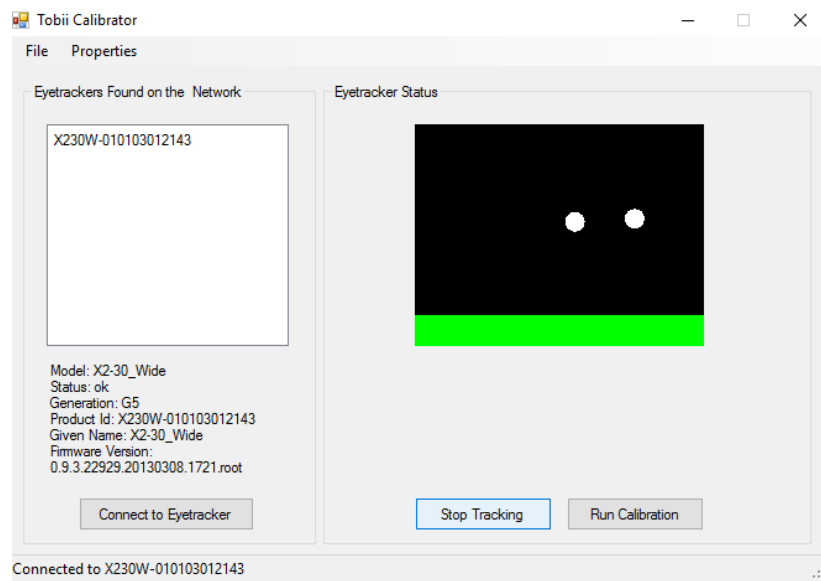


Figura 0-8: Tobii reconociendo ojos.

Una vez se haya estabilizado la mirada en el recuadro y la franja de color se mantenga en color verde, se puede proceder a realizar la calibración pulsando el botón “*Run Calibration*”. La prueba es muy similar a la de EyeTribe, teniendo que seguir el movimiento de una bola a lo largo de la pantalla. Al finalizar, los resultados los muestra en forma de desviación de las fijaciones. Cuanto menos se desvíen las líneas de los puntos, mejor habrán sido los resultados de la calibración. Esto es, si se muestran líneas largas saliendo de los círculos, es preciso realizar la calibración de nuevo. En la siguiente imagen se muestra una calibración prácticamente perfecta, la cual se usa como referencia para pasar los resultados de calibración a un sistema de puntuación de 5 estrellas.



Figura 0-9: Resultados calibración Tobii

E Código

```
public void InitRegistration(int audioDevice = 0)
{
    // Change format of the results folder
    ResultsFolder = ResultsFolder.Replace(@"\", @"\\");

    // Create a folder for user
    var userFolder = ResultsFolder + @"\\\" + UserId;

    // Check if folder_user exists
    if (!Directory.Exists(userFolder))
        Directory.CreateDirectory(userFolder);

    // Files to save reading pattern
    var fullReadingPattern = userFolder + @"\\full_reading_pattern.txt";
    var reducedReadingPattern = userFolder + @"\\reduced_reading_pattern.txt";

    FullFile = new StreamWriter(fullReadingPattern);
    ReducedFile = new StreamWriter(reducedReadingPattern);
    this.LeftEyeRegister = new List<Point>();
    this.RightEyeRegister = new List<Point>();

    // Start recording on the eye tracker has to be done before,
    // as it is done much slower than starting recording the audio
    // and can create a delay
    StartRecording();

    // audio file to be saved
    var audio = userFolder + @"\\audio.wav";

    AudioRecorder = new AudioRecorder(audio.Replace(@"\", @"\\"), audioDevice);
}
```

Figura 0-10: Código inicialización y comienzo del registro.

```
public void RegisterOcularActivity()
{
    if (this.TimeSpanMicrosec.Equals(this.TimeSpanPrev)) return;
    if (this.TimeSpanOriginal.Equals(-1)) this.TimeSpanOriginal = this.TimeSpanMilisec;

    this.TimeSpanPrev = this.TimeSpanMicrosec;

    var line = (this.TimeSpanMilisec - this.TimeSpanOriginal) + "\t" + IsFix + "\t" + State + "\t" + LeftEyeX
    + "\t" + (ScreenSizeY - LeftEyeY) + "\t" + LeftEyePSize + "\t" + LeftEyePCenterX + "\t" + (ScreenSizeY -
    LeftEyePCenterY) + "\t" + RightEyeX + "\t" + (ScreenSizeY - RightEyeY) + "\t" + RightEyePSize + "\t" +
    RightEyePCenterX + "\t" + (ScreenSizeY - RightEyePCenterY) + "\t";

    // Save all information in the full file.
    // Modify value of EyeY because origin coordinate is different.
    FullFile.WriteLine(line);

    // Save only the instants which there are values. In this case, the coordinate Y is also modified
    if (float.IsNaN(LeftEyeX) || float.IsNaN(LeftEyeY) || float.IsNaN(RightEyeX) || float.IsNaN(RightEyeY) ||
    Math.Abs(LeftEyeX) < 0 || Math.Abs(LeftEyeY) < 0 || Math.Abs(RightEyeX) < 0 || Math.Abs(RightEyeY) < 0 ||
    !(Math.Abs(LeftEyeX) > 0 && Math.Abs(LeftEyeX) < ScreenSizeX) ||
    !(Math.Abs(LeftEyeY) > 0 && Math.Abs(LeftEyeY) < ScreenSizeY) ||
    !(Math.Abs(RightEyeX) > 0 && Math.Abs(RightEyeX) < ScreenSizeX) ||
    !(Math.Abs(RightEyeY) > 0 && Math.Abs(RightEyeY) < ScreenSizeY)) return;

    ReducedFile.WriteLine(line);

    LeftEyeRegister.Add(new Point(LeftEyeX, LeftEyeY));
    RightEyeRegister.Add(new Point(RightEyeX, RightEyeY));
}
```

Figura 0-11: Código del registro de la actividad ocular I

```

// Break the loop when a key is pressed
private void OnKeyPress(object sender, KeyPressEventArgs keyPressEventHandler)
{
    _continue = false;
}

// Create thread and start recording
private void ComenzarRegistro_Click(object sender, EventArgs e)
{
    ComenzarRegistro.Hide();

    var thread = new Thread(RegistraActividad);
    thread.Start(SynchronizationContext.Current);
}

// Infinite loop for register the ocular activity until flag changes
private void RegistraActividad(object synchronizationContext)
{
    this.InitRegistro();

    while (_continue)
    {
        EyeTracker.RegisterOcularActivity();
    }

    this.FinalizarRegistro();

    ((SynchronizationContext)synchronizationContext).Send(CloseForm, null);
}

```

Figura 0-12: Código del registro de la actividad ocular II

```

public void OnGazeUpdate(GazeData gazeData)
{
    const int isTracking = GazeData.STATE_TRACKING_GAZE + GazeData.STATE_TRACKING_EYES +
        GazeData.STATE_TRACKING_PRESENCE;

    if (gazeData.State != isTracking) return;

    this.TimeSpanMilisec = gazeData.TimeStamp;
    this.TimeSpanMicrosec = gazeData.TimeStamp * 1000;

    // Is fix
    this.IsFix = gazeData.IsFixated;

    // State -> 32bit masked state integer
    this.State = gazeData.State;

    // Eyes
    var leftEye = gazeData.LeftEye;
    var rightEye = gazeData.RightEye;

    // Left
    this.LeftEyeX = leftEye.SmoothedCoordinates.X;
    this.LeftEyeY = leftEye.SmoothedCoordinates.Y;
    this.LeftEyePSize = leftEye.PupilSize;
    this.LeftEyePCenterX = leftEye.PupilCenterCoordinates.X;
    this.LeftEyePCenterY = leftEye.PupilCenterCoordinates.Y;

    // Right
    this.RightEyeX = rightEye.SmoothedCoordinates.X;
    this.RightEyeY = rightEye.SmoothedCoordinates.Y;
    this.RightEyePSize = rightEye.PupilSize;
    this.RightEyePCenterX = rightEye.PupilCenterCoordinates.X;
    this.RightEyePCenterY = rightEye.PupilCenterCoordinates.Y;
}

```

Figura 0-13: Obtención datos EyeTribe


```

public void OnGazeUpdate(object sender, GazeDataEventArgs gazeData)
{
    // Eyes
    var leftEye = gazeData.LeftEye;
    var rightEye = gazeData.RightEye;

    //TimeSpan
    this.TimeSpanMilisec = gazeData.DeviceTimeStamp / 1000;
    this.TimeSpanMicrosec = gazeData.DeviceTimeStamp;

    // Is fix
    this.IsFix = LeftEyeX.Equals(leftEye.GazePoint.PositionOnDisplayArea.X) &
        LeftEyeY.Equals(leftEye.GazePoint.PositionOnDisplayArea.Y) &
        RightEyeX.Equals(rightEye.GazePoint.PositionOnDisplayArea.X) &
        RightEyeY.Equals(rightEye.GazePoint.PositionOnDisplayArea.Y);

    // Left
    this.LeftEyeX = leftEye.GazePoint.PositionOnDisplayArea.X * ScreenSizeX;
    this.LeftEyeY = leftEye.GazePoint.PositionOnDisplayArea.Y * ScreenSizeY;
    this.LeftEyePSize = leftEye.Pupil.PupilDiameter;
    this.LeftEyePCenterX = 0;
    this.LeftEyePCenterY = 0;

    // Right
    this.RightEyeX = rightEye.GazePoint.PositionOnDisplayArea.X * ScreenSizeX;
    this.RightEyeY = rightEye.GazePoint.PositionOnDisplayArea.Y * ScreenSizeY;
    this.RightEyePSize = rightEye.Pupil.PupilDiameter;
    this.RightEyePCenterX = 0;
    this.RightEyePCenterY = 0;
}

```

Figura 0-14: Obtención datos Tobii

```

var backgroundPlot = "plot '" + readingFile + "' binary filetype=png origin=(0,0)
dx=" + propX + " dy=" + propY + " with rgbimage, ";

```

Figura 0-15: Fondo de los gráficos

```

plot.WriteLine("set xrange [0:" + xMax + "]");
plot.WriteLine("set yrange [0:" + yMax + "]");
plot.WriteLine("set title 'Left'");
plot.WriteLine("unset key"); // Hide legend
plot.WriteLine("set terminal png");
plot.WriteLine("set output '" + fullLeftPath);
plot.WriteLine(backgroundPlot + "'" + fullReadingPattern +
    "' using 4:5 with lines");

```

Figura 0-16: Código generación de resultados "Left_full.png" y "Right_full.png"

```

plot.WriteLine("set xrange [0:" + xMax + "]");
plot.WriteLine("set yrange [0:" + yMax + "]");
plot.WriteLine("set title 'Left'");
plot.WriteLine("unset key"); // Hide legend
plot.WriteLine("set terminal png");
plot.WriteLine("set output '" + fullLeftPath);
plot.WriteLine(backgroundPlot + "'" + fullReadingPattern +
    "' using 4:5 with lines");

```

Figura 0-17: Código generación de los resultados "left.png" y "right.png"

```

plot.WriteLine("set xrange [0:" + xMax + "]");
plot.WriteLine("set yrange [0:" + yMax + "]");
plot.WriteLine("set title 'Left with time'");
plot.WriteLine("unset key");
plot.WriteLine("set terminal png");
plot.WriteLine("set output '" + leftColor);
plot.WriteLine("set cbrange[0:" + totalTime + "]");
plot.WriteLine("set palette defined('" + div0 + "' 'red', '" + div1 + "' 'yellow', '" +
div2 + "' 'green', '" + totalTime + "' 'blue')");
plot.WriteLine(backgroundPlot + "'" + reducedReadingPattern + "' using 4:5:($1/1000)
with lines linecolor palette linewidth 1");

```

Figura 0-18: Código generación de los resultados "left_color.png" y "right_color.png"

```

plot.WriteLine("set xrange [0:'" + totalTime + "'");
plot.WriteLine("set yrange [0:200]");
plot.WriteLine("set title 'Left: Pupil size vs time'");
plot.WriteLine("unset key");
plot.WriteLine("set terminal png");
plot.WriteLine("set output '" + leftPupil);
plot.WriteLine("plot '" + reducedReadingPattern + "' using ($1/1000):6 with lines");

```

Figura 0-19: Código generación de los resultados "Left_pupil.png" y "Right_pupil.png"

```

for (var i = 0; i < nPhotograms; i++)
{
    var outFile = imagesLeft + i.ToString().PadLeft(nPhotograms.ToString().Length, '0') + ".png";
    plot.WriteLine(
        "set xrange [0:" + xMax + "]; " +
        "set yrange [0:" + yMax + "]; " +
        "unset key; " +
        "set title 'Left Eye'; " +
        "set output '" + outFile + "'; " +
        backgroundPlot + "''" + reducedReadingPattern +
        "' using 4:($1 <= " + (i * 1000) / frameRate + " ? $5 : 1/0) with lines;");
}

```

Figura 0-20: Código generación de los gráficos usados para los videos "Left_movie.mpeg" y "Right_movie.mpeg"

```

//Desplazamiento vertical hacia arriba
var maxLineRecorded = lineas.Max();
var minLineRecorded = lineas.Min();
var maxLine = Imagen.Lineas.Count - 1;
if (maxLineRecorded < maxLine)
{
    var i = 0;

    lineas = lineas.Select(x => x + maxLine - maxLineRecorded).ToList();

    foreach (var punto in puntos)
    {
        var c1 = Imagen.GetDistanceToLine(lineas[i], punto.X, punto.Y);
        distancias[i] = c1.Item2;
        i++;
    }
}

//Desplazamiento vertical hacia abajo
if (minLineRecorded > 0)
{
    var i = 0;

    lineas = lineas.Select(x => x - minLineRecorded).ToList();

    foreach (var punto in puntos)
    {
        var c1 = Imagen.GetDistanceToLine(lineas[i], punto.X, punto.Y);
        distancias[i] = c1.Item2;
        i++;
    }
}

```

Figura 0-21: Código comprobación de desplazamiento vertical

```

var primeraObservacionValida = (maxLineRecorded < maxLine) ? 0 : lineas.FindIndex(x => x == 0);
primeraObservacionValida = primeraObservacionValida == -1 ? 0 : primeraObservacionValida;

var ultimaObservacion = lineas.Count - primeraObservacionValida;

lineas = lineas.GetRange(primeraObservacionValida, ultimaObservacion);
distancias = distancias.GetRange(primeraObservacionValida, ultimaObservacion);
puntos = puntos.GetRange(primeraObservacionValida, ultimaObservacion);

```

Figura 0-22: Código limpieza de registros previos al comienzo de la lectura

```

for (var i = 1; i < puntos.Count; i++)
{
    var anteriorCoherente = coherencias.LastIndexOf(true);

    var lineaActual = lineas[i];
    var puntoActual = puntos[i].X;
    var ultimaLineaCoherente = lineas[anteriorCoherente];
    var ultimoPuntoCoherente = puntos[anteriorCoherente].X;

    var isRetroceso =
        (ultimaLineaCoherente == lineaActual && ultimoPuntoCoherente > puntoActual) ||
        (ultimaLineaCoherente - 1 == lineaActual && ultimoPuntoCoherente < puntoActual);

    var isAvance =
        (ultimaLineaCoherente == lineaActual && ultimoPuntoCoherente < puntoActual) ||
        (ultimaLineaCoherente + 1 == lineaActual && ultimoPuntoCoherente > puntoActual);

    var isIgual = ultimaLineaCoherente == lineaActual &&
        ultimoPuntoCoherente.Equals(puntoActual);

    var isCoherente = isAvance || isRetroceso || isIgual;

    coherencias.Add(isCoherente);

    // Para que se cuenten por bloques de retroceso y avance.
    if (retrocesos.LastOrDefault() != isRetroceso)
    {
        retrocesos.Add(isRetroceso);
    }

    if (isCoherente)
    {
        puntosCoherentes++;
    }
    else
    {
        var punto = puntos[i];
        var distToPrev =
            Imagen.GetDistanceToLine(ultimaLineaCoherente, punto.X, punto.Y);
        var distToNext = new Tuple<int, double>(-1, double.MaxValue);

        if (ultimaLineaCoherente + 1 < this.Imagen.GetNLineas())
        {
            distToNext =
                Imagen.GetDistanceToLine(ultimaLineaCoherente + 1, punto.X, punto.Y);
        }

        distancias[i] = Math.Min(distToPrev.Item2, distToNext.Item2);
    }

    distance += distancias[i];
}

var nRetrocesos = retrocesos.Count(x => x);

return new Precision
{
    DistanciaMediaFallo = distance / puntos.Count,
    PorcentajePosicionesCoherentes = (double) (puntosCoherentes * 100) / puntos.Count,
    Retrocesos = nRetrocesos
};

```

Figura 0-23: Código cálculo de la precisión

F Cálculo de la precisión y la distancia media de fallo

Para realizar el cálculo de la precisión y la distancia media de fallo se ha ideado el siguiente razonamiento. Si se tiene un texto como el de la figura 0-24, mediante el uso de un programa de edición de imágenes (como Gimp o Photoshop) se puede saber la posición de las líneas sacando cuatro puntos de referencia por línea.

Había una vez una niña muy linda que vivía en el bosque con su mamá, que le había hecho una capa roja para protegerse del frío y el viento. A la niña le gustaba tanto la capa que la llevaba a todas horas, por lo que todo el mundo la llamaba Caperucita Roja.

Un día, su abuelita que vivía al otro lado del bosque se puso malita y su madre le pidió que le llevase unos pasteles, frutas y miel.

- Querida hijita, llévale estos alimentos a la abuelita y sobre todo no te apartes del camino, ya que en el bosque hay lobos y es muy peligroso - le dijo

Caperucita Roja recogió la cesta con los pasteles, la fruta y la miel y se puso en camino.

Figura 0-24: Texto de ejemplo

Esto da para cada línea un rectángulo, tal y como se muestra en la figura 0-25, teniendo por vértices 4 puntos, haciendo que las alturas de todas las líneas sean iguales, y estén centradas respecto al centro de la línea y que el ancho de estas venga limitado por la primera y la última letra de cada línea.

Había una vez una niña muy linda que vivía en el bosque con su mamá, que le
había hecho una capa roja para protegerse del frío y el viento. A la niña le
gustaba tanto la capa que la llevaba a todas horas, por lo que todo el mundo
la llamaba Caperucita Roja.
Un día, su abuelita que vivía al otro lado del bosque se puso malita y su madre
le pidió que le llevase unos pasteles, frutas y miel.
- Querida hijita, llévale estos alimentos a la abuelita y sobre todo no te apartes
del camino, ya que en el bosque hay lobos y es muy peligroso - le dijo
Caperucita Roja recogió la cesta con los pasteles, la fruta y la miel y se puso
en camino.

Figura 0-25: Texto con las líneas delimitadas

Se anotan las posiciones de los vértices de estos rectángulos en forma de tanto por uno en un fichero csv. El que sea en tanto por uno es para que sea independiente de las dimensiones de la pantalla y se pueda usar con cualquier tipo de resolución y es el programa el que se encarga de devolverlo a distancias en píxeles.

Después, se calcula la distancia de cada punto a las distintas líneas usando la fórmula de la distancia euclídea, adaptando las distancias entre el punto y el rectángulo teniendo en cuenta las siguientes consideraciones (fijándonos en la figura 0-26):

1. Si la coordenada X está situada entre X_0 y X_2 , o, lo que es lo mismo, entre X_1 y X_3 , la distancia respecto al eje X es 0. En caso contrario, es la menor de las distancias a X_0 y X_2 .
2. Si la coordenada Y está situada entre Y_0 y Y_1 , o, lo que es lo mismo, entre Y_2 e Y_3 , la distancia respecto al eje Y es 0. En caso contrario, es la menor de las distancias a Y_0 e Y_1 .

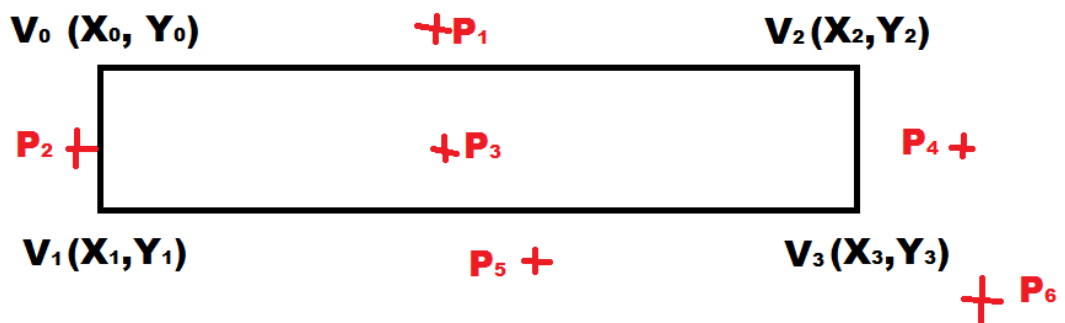


Figura 0-26: Cálculo de la distancia

Con estos criterios, la distancia del punto P_1 respecto al eje X, sería 0, ya que está entre X_0 y X_2 y la distancia respecto al eje Y sería la distancia de P_1 a la línea que unen V_0 y V_2 . La distancia del punto P_2 respecto al eje Y, sería 0, ya que está entre Y_0 e Y_1 y la distancia respecto al eje X sería la distancia de P_2 a la línea que unen V_0 y V_1 . La distancia del punto P_3 respecto al eje X, sería 0, ya que está entre X_0 y X_2 , al igual que la distancia respecto al eje Y, por estar entre Y_0 e Y_1 . La distancia del punto P_4 respecto al eje Y, sería 0, ya que está entre Y_0 e Y_1 y la distancia respecto al eje X sería la distancia de P_4 a la línea que unen V_2 y V_3 . La distancia del punto P_5 respecto al eje X, sería 0, ya que está entre X_0 y X_2 y la distancia respecto al eje Y sería la distancia de P_5 a la línea que unen V_1 y V_3 . La distancia del punto P_6 al rectángulo, es la misma que la distancia al vértice V_3 .

Las distancias respecto a los distintos ejes se incorporan a la ecuación de la distancia euclídea.

$$d_E(P_1, P_2) = \sqrt{(distancia\ eje\ X)^2 + (distancia\ eje\ Y)^2}$$

Conociendo estas distancias, se clasifican los puntos en las líneas, asignándoles la línea más cercana a cada punto, independientemente de que sea coherente en la sucesión de los puntos. Si un punto está a la misma distancia de dos líneas, se toma la primera de ellas. Por ejemplo, en la figura 0-27 tendríamos que los puntos P_0 , P_1 , P_2 , P_3 y P_{n-1} pertenecen a la línea 1, los

puntos P_n , P_{n+1} y P_{n+3} pertenecen a la línea 2 y el punto P_{n+2} pertenece, inicialmente, a la línea 3.

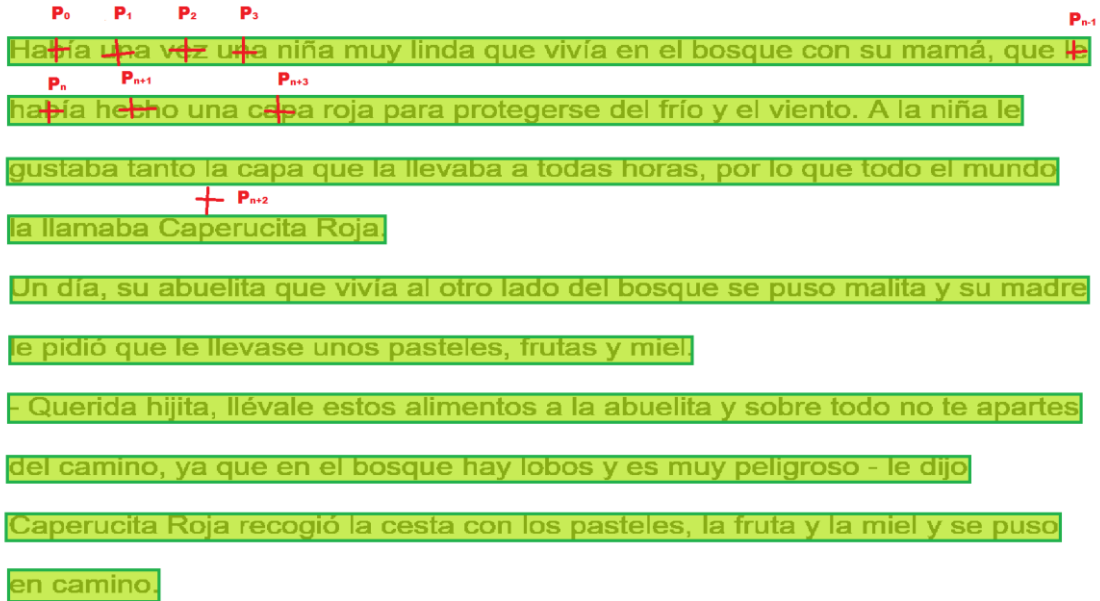


Figura 0-27: Texto con líneas y puntos marcados

Puesto que en la prueba se han de leer todas las líneas del texto, se puede asegurar que siempre se ha de empezar en la primera línea y terminar en la última. Por ello, se sabe si hubo algún tipo de desplazamiento vertical que afecte a la precisión comprobando cuáles son la primera y la última líneas leídas y ajustando las líneas más cercanas, en función del número de líneas que se hayan desplazado.

Para calcular la precisión se ha de definir lo que son los puntos coherentes. Estos son aquellos puntos que suponen un avance, retroceso o fijación. Un avance se da cuando, estando en la misma línea, aumenta la posición del eje x, o, al pasar a la línea siguiente, se retrocede en el eje x. Un retroceso es justo lo opuesto; es decir, estando en la misma línea, se pasa a una posición anterior del eje x, o, al pasar a la línea anterior, se avanza en el eje x. Finalmente, la fijación se produce cuando se mantiene en la misma línea y posición del eje x. En todos los casos se hace la comparación con el anterior punto coherente.

Para los puntos no coherentes (como el caso del punto P_{n+2} de la figura 0-27), se calcula la distancia a la anterior línea coherente y a la siguiente a esta y se utiliza el menor de los valores para el cálculo del error correspondiente a este punto.

Finalmente, se obtiene el porcentaje de posiciones coherentes dividiendo el número de puntos coherentes respecto al número total de puntos registrados. Y la distancia media de fallo se calcula como la media de las distancias.